



TUGAS AKHIR -TE 141599

ANALISIS PERBANDINGAN *SUPERVISORY PREDICTIVE CONTROL* (SPC) UNTUK KONFIGURASI *CASCADE* DAN *PARALLEL* PADA PERFORMANSI SISTEM PENGATURAN PROSES DENGAN VARIABEL WAKTU TUNDA

Riki Rizki
NRP 2212 100 017

Dosen Pembimbing
Ir. Ali Fatoni, MT.
Imam Arifin, ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT -TE 141599

**COMPARATIVE ANALYSIS SUPERVISORY PREDICTIVE
CONTROL (SPC) FOR CASCADE AND PARALLEL
CONFIGURATION IN PROCESS CONTROL SYSTEM WITH
VARIABLE TIME DELAY**

Riki Rizki
NRP 2212 100 017

Supervisor
Ir. Ali Fatoni, MT.
Imam Arifin, ST., MT.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2016

**ANALISIS PERBANDINGAN SUPERVISORY PREDICTIVE
CONTROL (SPC) UNTUK KONFIGURASI CASCADE DAN
PARALLEL PADA PERFORMANSI SISTEM PENGATURAN
PROSES DENGAN VARIABEL WAKTU TUNDA**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

Pada

Bidang Studi Teknik Sistem Pengaturan

Jurusan Teknik Elektro

Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Ali Fatoni, MT.

NIP. 196206031989031002

Imam Arifin, ST., MT.

NIP. 197302222002121001



PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Analisa Perbandingan Konfigurasi Cascade dan Parallel Supervisory Predictive Control (SPC) pada Performansi Sistem Pengaturan Proses dengan Variabel Waktu Tunda**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2016

Riki Rizki
NRP 2212100017

ANALISIS PERBANDINGAN *SUPERVISORY PREDICTIVE CONTROL (SPC)* UNTUK KONFIGURASI *CASCADE* DAN *PARALLEL* PADA PERFORMANSI SISTEM PENGATURAN PROSES DENGAN VARIABEL WAKTU TUNDA

Nama : Riki Rizki
Pembimbing 1 : Ir. Ali Fatoni, MT.
Pembimbing 2 : Iman Arifin, ST., MT.

ABSTRAK

Banyak sistem yang memiliki fenomena waktu tunda (*time delay*) dalam dinamikanya. Pada sistem pengaturan proses waktu tunda sering terjadi yang disebabkan oleh bentuk fisik *plant*, aksi kontroler dan operasi aktuator. Sistem yang mengalami waktu tunda dapat menyebabkan sistem tersebut menjadi *time varying*. Adanya waktu tunda didalam suatu sistem kontrol umpan balik akan menjadi kendala yang serius dalam mencapai performansi yang baik. Kontrol prediktif merupakan metode kontrol yang populer dalam menangani sistem dengan dinamika yang lambat. Penerapan metode ini dapat dilakukan dalam dua level yaitu *regulatory* dan *supervisory*. Pada penelitian ini, akan dibandingkan penerapan dari kontroler PI dan prediktif pada level *regulatory* dan *supervisory* dengan konfigurasi *cascade* dan *parallel*. Penerapan dari keempat metode tersebut akan dilakukan pada sistem pengaturan level PCT-100. Penerapan sistem kontrol prediktif ini mampu menghilangkan efek dari waktu tunda dan memperbaiki performansi sistem dengan mempercepat respon. Konfigurasi *cascade* pada kontroler prediktif yang berada pada level *supervisory* mendapatkan hasil yang paling bagus yaitu memiliki konstanta waktu paling kecil sebesar 1.026 detik saat pengujian perubahan *set point*.

Kata Kunci: *supervisory, regulatory, predictive control, time varying, cascade, parallel*

COMPARATIVE ANALYSIS SUPERVISORY PREDICTIVE CONTROL (SPC) FOR CASCADE AND PARALLEL CONFIGURATION IN PROCESS CONTROL SYSTEM WITH VARIABLE TIME DELAY

Name : Riki Rizki
Supervisor 1 : Ir. Ali Fatoni, MT.
Supervisor 2 : Iman Arifin, ST., MT.

ABSTRACT

Many processes include time delay phenomena in their inner dynamics. In process control system time delay usually happen because measuring, controller action, and actuator operation, which are possibly time-varying. Thus there is an increasing interest in studying time delay systems in all scientific areas, especially in control engineering. Predictive control is a popular technique to control slow dynamical systems. This controller can applied in two level that are regulatory and supervisory. In this research, comparison between PI and Predictive controller in level regulatory and supervisory with cascade and parallel configuration will analyze. Implementation of this control method will be applied to level control systems PCT-100. Predictive controller can eliminate the impact of time delay and improve system performance with fast time constant. Predictive control in supervisory level with cascade configuration has the best response with the smallest time constant 1.206 second when set point variation test.

Keywords: supervisory, regulatory, predictive control, time varying, cascade, parallel

KATA PENGANTAR

Segala puji dan syukur penulis lantunkan atas kehadrirat Allah SWT karena atas segala rahmat, taufik, dan hidayah-Nya sehingga penelitian ini dapat terselesaikan dengan baik. Shalawat dan salam juga penulis sampaikan kepada Rasulullah Muhammad SAW yang telah menjadi teladan bagi penulis.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Strata-1 pada Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

ANALISIS PERBANDINGAN *SUPERVISORY PREDICTIVE CONTROL* (SPC) UNTUK KONFIGURASI *CASCADE* DAN *PARALLEL* PADA PERFORMANSI SISTEM PENGATURAN PROSES DENGAN VARIABEL WAKTU TUNDA

Dalam pengerjaan penelitian ini, penulis mendapat banyak sekali bantuan dan dukungan dari berbagai pihak. Oleh karena itu, dengan segala hormat, penulis mengucapkan terima kasih kepada Bapak Ali Fatoni dan Bapak Imam Arifin selaku dosen pembimbing yang telah membimbing dengan penuh kesabaran, orang tua yang selalu mendoakan, seluruh keluarga, warga Lab AJ.104, teman-teman bidang studi Teknik Sistem Pengaturan dan semua pihak yang memberikan dukungan penuh kepada penulis dalam penyelesaian penelitian ini.

Penulis menyadari bahwa pengerjaan penelitian ini masih jauh dari sempurna. Untuk itu, penulis memohon maaf atas segala kekurangan dalam laporan penelitian ini. Semoga penelitian ini dapat memberikan manfaat dalam pengembangan ilmu pengetahuan dan teknologi.

Surabaya, Januari 2016

Penulis

DAFTAR ISI

PERNYATAAN KEASLIAN	iii
LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Permasalahan	2
1.3. Batasan Masalah	2
1.4. Tujuan	2
1.5. Metodologi	2
1.6. Sistematika Penulisan	4
1.7. Relevansi	5
BAB 2 SISTEM PENGATURAN PROSES DAN KONTROLER	7
2.1. Sistem Kontrol Proses	7
2.2. Identifikasi Sistem	9
2.2.1. Identifikasi Umpan Balik Tertutup	9
2.2.2. Identifikasi Metode Strejc	10
2.3. Fungsi Alih dengan Waktu Tunda	13
2.4. Process Control Technology (PCT-100)[12]	14
2.5. Pemodelan Sistem	15
2.5.1. Perhitungan Waktu tunda	16
2.5.2. Pemodelan Tanki	17
2.5.3. Pendekatan <i>Deret Taylor</i>	19
2.6. Kontroler PID	19
2.6.1. Kontroler P (<i>Proportional Controller</i>)	20
2.6.2. Kontroler PI (<i>Proportional + Integral Controller</i>)	20
2.6.3. Kontroler PD (<i>Proportional + Derivative Controller</i>)	21
2.6.4. Kontroler PID (<i>Proportional + Integral + Derivative Controller</i>)	21

2.7.	<i>Model Predictive Control (MPC)</i>	22
2.7.1.	Penurunan Persamaan Kontroler Prediktif.....	22
2.7.2	Sistem Pengaturan Umpan Balik Tertutup.....	27
2.8.	<i>Supervisory Predictive Control (SPC)</i>	29
2.8.1	Konfigurasi <i>Cascade</i> SPC	29
2.8.2	Konfigurasi <i>Parallel</i> SPC.....	32
2.8.3	Penurunan Rumus SPC	34
2.9	<i>Root Mean Square Error (RMSE)</i>	37
BAB 3	PERANCANGAN SISTEM	39
3.1.	Gambaran Umum Sistem	39
3.2.	Identifikasi Sistem.....	40
3.2.1.	Validasi Model	43
3.2.2.	Model Umpan Balik Terbuka.....	44
3.3.	Model Matematis Sistem.....	45
3.2.1	Mencari Nilai Parameter Sistem Saat Titik Kerja	45
3.2.2	Menghitung Fungsi Alih Sistem.....	47
3.2.3	Pendekatan Deret Taylor	47
3.4.	Pengujian Respon Umpan Balik Tertutup Tanpa Kontroler	48
3.5.	Perancangan Kontroler PI	49
3.6.	Perancangan Kontroler MPC.....	51
3.7.	Perancangan SPC	53
3.8.	Kestabilan Sistem dengan Waktu Tunda.....	56
BAB 4	PENGUJIAN DAN ANALISIS	61
4.1.	Simulasi Kontroler PI dan Analisis	61
4.1.1.	Simulasi dengan Masukan Sinyal <i>Step</i>	61
4.1.2.	Simulasi dengan Variasi <i>Set Point</i>	62
4.2.	Simulasi Kontroler MPC Level <i>Regulatory</i> dan Analisis	63
4.2.1.	Simulasi dengan Masukan Sinyal <i>Step</i>	63
4.2.2.	Simulasi MPC dengan Variasi <i>Set Point</i>	64
4.2.3.	Simulasi dengan Variasi <i>Np</i>	65
4.3.	Simulasi Kontroler SPC Konfigurasi <i>Cascade</i> dan Analisis.....	66
4.3.1.	Simulasi dengan Masukan Sinyal <i>Step</i>	66

4.3.2.	Simulasi dengan Variasi <i>Set Point</i>	68
4.3.3.	Simulasi dengan Variasi <i>Np</i>	68
4.4.	Simulasi Kontroler SPC Konfigurasi <i>Parallel</i> dan Analisis	69
4.4.1.	Simulasi dengan Masukan Sinyal <i>Step</i>	69
4.4.2.	Simulasi dengan Variasi <i>Set Point</i>	70
4.4.3.	Simulasi dengan Variasi <i>Np</i>	71
4.5.	Perbandingan Kontroler PI dan Prediktif	72
4.5.1.	Perbandingan Kontroler PI dan Prediktif Saat Uji Coba <i>Set Point</i> Sinyal <i>Step</i>	72
4.5.2.	Perbandingan Kontroler PI dan Prediktif Saat Uji Coba <i>Set Point Tracking</i>	73
4.6.	Implementasi	74
4.6.1.	Kontroler PI	75
4.6.2.	Kontroler MPC	76
BAB 5 PENUTUP		79
5.1.	Kesimpulan	79
5.2.	Saran	79
DAFTAR PUSTAKA		81
LAMPIRAN		83
RIWAYAT HIDUP		113

Table of Content

AUTENTICITY	iii
APPROVAL SHEET	v
<i>ABSTRAK</i>	vii
ABSTRACT	ix
PREFACE	xi
TABLE OF CONTENT	xiii
ILLUSTRATION	xvii
TABLE	xix
CHAPTER I INTRODUCTION	1
1.1. Background	1
1.2. Problems	2
1.3. Problems Limitation	2
1.4. Research Objective	2
1.5. Methodology	2
1.6. Writing Systematic	4
1.7. Relevance	5
BAB 2 PROCESS CONTROL SYSTEM AND CONTROLLER	7
2.1. Process Control System	7
2.2. System Identification	9
2.2.1. Close Loop Identification	9
2.2.2. Strejc Method Identification	10
2.3. Transfer Function with Delay Time	13
2.4. <i>Process Control Technology</i> (PCT-100)	14
2.5. System Modelling	15
2.5.1. Delay Time Computation	16
2.5.2. Tank Modelling	17
2.5.3. Deret Taylor Approximation	19
2.6. PID Controller	19
2.6.1. Proportional Controller	20
2.6.2. Proportional + Integral Controller	20
2.6.3. Proportional + Derivative Controller	21
2.6.4. Proportional + Integral + Derivative Controller	21
2.7. Model Predictive Control (MPC)	22
2.7.1. Predictive Control Equation	22

2.7.2.	Closed Loop Control Systems	27
2.8.	Supervisory Predictive Control (SPC).....	29
2.8.1.	Cascade SPC Configuration	29
2.8.2.	Parallel SPC Configuration	32
2.8.3.	SPC Equation	34
2.9.	Root Mean Square Error (RMSE)	37
BAB 3	SYSTEM DESIGN	39
3.1.	System Overview	39
3.2.	System Identification.....	40
3.2.1.	Model Validation.....	43
3.2.2.	Open Loop Model	44
3.3.	Mathematical Model of System.....	45
3.3.1	System Parameter in Operation Point.....	45
3.3.2	Transfer Function of System	47
3.3.3	Deret Taylor Approximation	47
3.4.	Closed Loop Test without Controller	48
3.5.	PI Controller Design.....	49
3.6.	MPC Controller Design.....	51
3.7.	SPC Design	53
3.8.	System Stability with Delay Time.....	56
BAB 4	TESTING AND ANALYSIS.....	61
4.1.	PI Controller Simulation and Analysis.....	61
4.1.1.	Simulation with Input Step Signal.....	61
4.1.2.	Simulation with setpoint variation.....	62
4.2.	MPC Level <i>Regulatory</i> Simulation and Analysis	63
4.2.1.	Simulation with Input Step Signal.....	63
4.2.2.	MPC Simulation with setpoint variation	64
4.2.3.	Simulation with Np Variation	65
4.3.	Cascade Configuration SPC Simulation and Analysis...66	
4.3.1.	Simulation with Input Step Signal.....	66
4.3.2.	Simulation with setpoint variation.....	68
4.3.3.	Simulation with Np Variation	68
4.4.	Parallel Configuration SPC Simulation and Analysis69	
4.4.1.	Simulation with Input Step Signal.....	69
4.4.2.	Simulation with setpoint variation.....	70

4.4.3.	Simulation with Np Variation.....	71
4.5.	PI and Predictive comparison.....	72
4.5.1.	PI and Predictive comparison with input step signal	72
4.5.2.	PI and Predictive comparison with setpoint variation...	73
4.6.	Implementation.....	74
4.6.1.	PI Controller.....	75
4.6.2.	MPC Controller	76
BAB 5	CLOSING.....	79
5.1.	Conclusion.....	79
5.2.	Suggestion	79
	BIBLIOGRAPHY	81
	ENCLOSURE	83
	BIOGRAPHY	113

DAFTAR GAMBAR

Gambar 2. 1.	Diagram blok sistem kontrol	7
Gambar 2. 2.	Spesifikasi respon orde satu.....	8
Gambar 2. 3.	Spesifikasi respon orde 2.....	9
Gambar 2. 4.	Diagram blok umpan balik terbuka	10
Gambar 2. 5.	Kurva S metode strejc.....	10
Gambar 2. 6.	<i>Process control technology</i> [12].....	14
Gambar 2. 7.	Representasi sistem tanki.....	16
Gambar 2. 8.	Metode <i>model predictive control</i>	25
Gambar 2. 9.	Blok diagram <i>predictive control</i>	28
Gambar 2. 10.	Konfigurasi SPC	29
Gambar 2. 11.	Konfigurasi <i>cascade</i> SPC	30
Gambar 2. 12.	Konfigurasi <i>parallel</i> SPC	33
Gambar 3. 1.	Arsitektur sistem SCADA	39
Gambar 3. 2.	Diagram blok identifikasi	40
Gambar 3. 3.	Penarikan garis strejc pada respon.....	41
Gambar 3. 4.	Validasi model	43
Gambar 3. 5.	Diagram pemodelan sistem.....	45
Gambar 3. 6.	Respon umpan balik terbuka sistem	48
Gambar 3. 7.	Flowchart dari kontroler MPC	52
Gambar 3. 8.	Flowchart algoritma SPC.....	55
Gambar 3. 9.	Diagram bode fungsi alih model 1.....	57
Gambar 3. 10.	Diagram bode fungsi alih model 2.....	57
Gambar 3. 11.	Nyquist fungsi alih model 1.....	58
Gambar 3. 12.	Nyquist fungsi alih model 2.....	59
Gambar 4. 1.	Diagram blok simulasi PI	61
Gambar 4. 2.	Respon sistem dengan kontroler PI	62
Gambar 4. 3.	Respon kontroler PI terhadap masukan variasi <i>set point</i>	63
Gambar 4. 4.	Blok diagram kontroler prediktif.....	63
Gambar 4. 5.	Respon sistem dengan kontroler MPC diberi masukan sinyal <i>step</i>	64
Gambar 4. 6.	Pengujian kontroler MPC dengan variasi set point	65

Gambar 4. 7. Respon sistem dengan kontroler MPC variasi N_p	66
Gambar 4. 8. Respon sistem dengan kontroler SPC konfigurasi <i>cascade</i> diberi masukan sinyal <i>step</i>	67
Gambar 4. 9. Respon sistem dengan kontroler SPC konfigurasi <i>cascade</i> dengan variasi <i>set point</i>	68
Gambar 4. 10. Respon sistem dengan kontroler SPC konfigurasi <i>cascade</i> dengan variasi N_p	69
Gambar 4. 11. Respon sistem dengan kontroler SPC konfigurasi <i>parallel</i> diberi masukan sinyal <i>step</i>	70
Gambar 4. 12. Respon sistem dengan kontroler SPC konfigurasi <i>parallel</i> dengan variasi <i>set point</i>	71
Gambar 4. 13. Respon sistem dengan kontroler SPC konfigurasi <i>parallel</i> dengan variasi N_p	72
Gambar 4. 14. Perbandingan respon dengan masukan sinyal <i>step</i>	73
Gambar 4. 15. Perbandingan respon sistem dengan kontroler yang berbeda	74
Gambar 4. 16. Skema implementasi	74
Gambar 4. 17. Susunan skema implementasi berdasarkan <i>hardware</i>	75
Gambar 4. 18. Respon implementasi kontroler PI	76
Gambar 4. 19. Kontroler MPC saat implementasi	77

DAFTAR TABEL

Tabel 2. 1. Parameter Metode Strejc	11
Tabel 2. 2. Parameter sistem.....	15
Tabel 2. 3. Perbandingan parameter k_p , k_i , dan k_d	20
Tabel 3. 1. Model matematis umpan balik tertutup	42
Tabel 3. 2. Perbandingan nilai RMSE tiap model matematis	44
Tabel 3. 3. Parameter pendekatan SOPDT	49
Tabel 3. 4. Parameter kontroler PI.....	50
Tabel 4. 1. Perbandingan respon kontroler PI	61
Tabel 4. 2. Perbandingan respon kontroler MPC	64
Tabel 4. 3. Perbandingan respon kontroler SPC <i>cascade</i>	67
Tabel 4. 4. Perbandingan parameter respon pada kontroler SPC <i>parallel</i>	70

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Dalam perkembangan dunia yang modern, sektor industri dihadapkan pada tantangan baru mengenai peningkatan produksi dan penurunan biaya produksi. Sistem *Supervisory Control and Data Acquisition* (SCADA) telah banyak digunakan untuk mengatasi penyebaran informasi dan data dalam suatu sistem otomasi industri. Dalam pengembangannya *supervisory control* telah mengalami peningkatan dalam hal kecepatan pada penyebaran informasi dan data dalam suatu sistem SCADA. Kecepatan pemrosesan yang cepat akan digunakan untuk optimasi sistem. Telah banyak algoritma yang diterapkan dalam *supervisory control*, salah satunya adalah *supervisory predictive control* (SPC) [1]. Terdapat dua bentuk konfigurasi yang sering digunakan dalam penerapan SPC yaitu konfigurasi *cascade* dan *parallel* [2].

Model Predictive Control (MPC) memiliki sejarah yang panjang dalam ruang lingkup teknik sistem pengaturan. Konsep ini menjadi salah satu yang paling diminati para peneliti dalam dunia akademik maupun di industri. Objektif kontrol dari penerapan MPC adalah menghitung sinyal kontrol yang akan datang untuk mengoptimasi sinyal keluaran pada waktu selanjutnya. Jenis kontroler ini telah banyak digunakan akhir-akhir ini, dan diterima sebagai perangkat penting pada aplikasi dalam proses industri [3].

Banyak sistem yang memiliki fenomena waktu tunda (*time delay*) dalam dinamikanya, contohnya seperti pada bidang biologi, kimia, fisika, mekanik, dan teknik. Untuk tambahan pada aktuator, sensor, *field network* yang menerapkan sistem kontrol *feedback* akan mengalami waktu tunda yang menyebabkan sistem tersebut menjadi *time varying*. Oleh karena itu, permasalahan waktu tunda ini menjadi suatu topik yang menarik untuk dibahas dibidang teknik, khususnya di bidang teknik sistem pengaturan [4]. Adanya waktu tunda didalam suatu sistem kontrol umpan balik akan menjadi kendala yang serius dalam mencapai performansi yang baik.

Implementasi dari SPC pada sektor industri telah banyak berhasil diterapkan seperti yang dilaporkan pada [2], [5], [6], dan [7]. Pada penelitian ini akan dilakukan analisis perbandingan terhadap dua konfigurasi dari SPC yaitu *cascade* dan *parallel* dalam menghilangkan

efek dari waktu tunda yang terjadi pada sistem pengaturan proses. Penerapan SPC akan dibandingkan dengan kontroler prediktif pada level *regulatory* dan kontroler PI.

1.2. Permasalahan

Dalam suatu sistem pengaturan proses masalah yang sering muncul adalah waktu tunda. Waktu tunda yang terjadi diakibatkan oleh beberapa hal yaitu waktu yang dibutuhkan dalam menyalurkan energi atau informasi, akumulasi dari keteringgalan waktu (*time lags*), dan waktu yang dibutuhkan kontroler dalam memproses algoritma yang kompleks[8].

Keterlambatan yang muncul akibat adanya waktu tunda akan berdampak juga pada kinerja dari aktuator dan sensor yang digunakan. Adanya waktu tunda didalam suatu sistem kontrol umpan balik akan menjadi suatu kendala yang serius dalam mencapai performansi yang baik yaitu tidak mampu mencapai kriteria yang diinginkan seperti *time constant*, *rise time*, dan *settling time*.

1.3. Batasan Masalah

Implementasi dari kontroler yang digunakan adalah pada *Process Control Technology* (PCT-100) dengan variabel proses yang diamati adalah level. Pendekatan waktu tunda menggunakan pendekatan Taylor orde pertama.

1.4. Tujuan

Pada penelitian ini akan dilakukan perbandingan antara kontroler MPC yang digunakan pada level *regulatory* dan *supervisory* dengan konfigurasi *cascade* dan *parallel*. Penerapan algoritma prediktif akan digunakan untuk menghilangkan dampak waktu tunda dalam sistem pengaturan proses, sehingga dapat meningkatkan performansi sistem.

1.5. Metodologi

Prosedur penelitian yang dilakukan dalam penelitian ini adalah sebagai berikut:

a. Studi literatur

Tahap ini dilakukan untuk mengetahui prinsip dari alat dan metode yang digunakan dalam penelitian ini. Pada tahapan ini pekerjaan akan dibagi menjadi dua bagian yaitu:

- Pengumpulan data/informasi
Pencarian dan pengumpulan literatur dan buku-buku yang berkaitan dengan penelitian ini, serta *manual book* dari peralatan yang digunakan.
- Observasi
Pengamatan terhadap karakteristik dari peralatan yaitu cara penggunaan, cara kerja, dan cara mengontrolnya.

b. Identifikasi *plant* dan pemodelan

Sebelum melakukan identifikasi *plant* terlebih dahulu dilakukan pemodelan dengan penurunan rumus fisika untuk keperluan simulasi. Identifikasi sistem dilakukan dalam keadaan umpan balik tertutup dengan memberikan sinyal uji berupa *unity step*. Metode pendekatan yang digunakan adalah *Strejc*. Melalui analisis dengan menggunakan tabel *Strejc* maka akan didapatkan fungsi alih dari *plant*.

c. Perancangan Kontroler

Pada bagian *supervisory layer* akan dirancang algoritma dari *Model Predictive Control* (MPC) dengan dua konfigurasi *cascade* dan *parallel*. Sementara pada bagian *regulatory layer* akan dirancang kontroler MPC dan kontroler PI dengan perhitungan gain (K_p dan T_i) menggunakan metode *Internal Model Control* (IMC) [9].

d. Simulasi

Hasil pemodelan sistem dan perancangan kontroler akan disimulasikan terlebih dahulu sebelum diimplementasikan dengan menggunakan bantuan *software*. Pada simulasi yang dilakukan sistem juga akan diuji dengan variasi *set point* dalam waktu yang berkelanjutan.

e. Implementasi

Hasil percobaan yang telah disimulasi akan diimplementasikan pada PCT-100. Data yang diperoleh akan dianalisis berkaitan dengan performansi sistem setelah diberi kontroler. Dari hasil ini akan didapatkan kesimpulan dari penelitian ini.

f. Penyusunan Laporan Penelitian

Penulisan laporan ini berupa laporan ilmiah yang mencakup semua proses pengerjaan penelitian, mulai dari pendahuluan, tinjauan pustaka,

perancangan sistem, simulasi dan implementasi, serta penutup yang berisi kesimpulan dan saran yang diperoleh dalam penelitian ini.

1.6. Sistematika Penulisan

Sistematika penulisan laporan penelitian ini dapat dijelaskan sebagai berikut :

BAB I : PENDAHULUAN

Dalam Bab I akan dijelaskan mengenai latar belakang, permasalahan, batasan masalah, tujuan, metodologi, dan sistematika penulisan laporan penelitian ini.

BAB II : SISTEM PENGATURAN PROSES DAN KONTROLER

Penjelasan mengenai konsep-konsep yang mendasari perancangan penelitian ini meliputi, sistem pengaturan proses, *model predictive control*, *supervisory control*, dan studi literatur lainnya dijelaskan pada bagian ini.

BAB III : PERANCANGAN SISTEM

Tahapan dalam perancangan sistem seperti pemodelan sistem, komponen-komponen sistem, dan perancangan algoritma kontrol yang akan diimplementasikan dijabarkan pada Bab III.

BAB IV : IMPLEMENTASI DAN ANALISIS

Bab VI akan membahas tentang hasil simulasi dan implementasi dari kontroler pada sistem pengaturan level PCT-100. Analisis performansi sistem terhadap sinyal masukan yang bervariasi dalam waktu yang berkelanjutan.

BAB V : KESIMPULAN DAN SARAN

Pada bagian terakhir ini dijelaskan mengenai kesimpulan dan saran – saran berkaitan dengan pengembangan penelitian.

1.7. Relevansi

Pemanfaatan kontroler pada level *supervisory* telah banyak dikembangkan akhir-akhir ini, sehingga penelitian ini akan menjadi referensi dalam penerapan *supervisory* kontrol dengan metode kontrol prediktif. Pengujian yang dilakukan pada penelitian ini juga berdasarkan kondisi riil yang terjadi di industri yaitu berupa perubahan *set point* yang dilakukan karena beberapa kondisi misalnya peningkatan produksi.

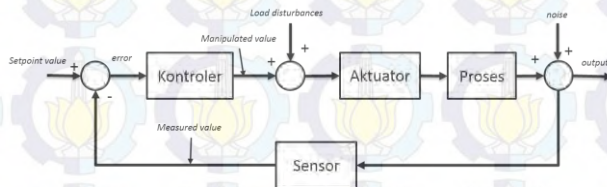
BAB 2

SISTEM PENGATURAN PROSES DAN KONTROLER

Bab ini membahas mengenai sistem pengaturan proses beserta komponennya, pengertian dari kontroler yang digunakan serta penurunan rumus.

2.1. Sistem Kontrol Proses

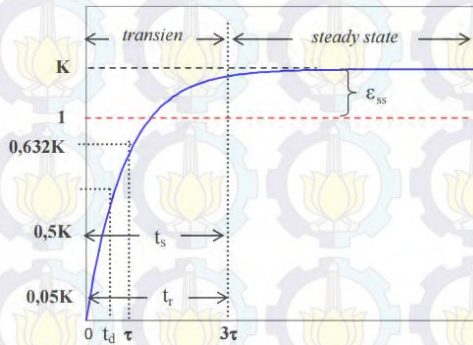
Sistem kontrol proses adalah gabungan kerja dari alat-alat pengendalian otomatis. Semua peralatan yang membentuk sistem kontrol disebut instrumentasi kontrol proses. Dalam analisis sistem kontrol selalu diawali dengan tampilan diagram blok sistem seperti terlihat pada Gambar 2. 1. Pada diagram blok ini akan menunjukkan elemen yang berperan dalam sistem tersebut serta proses yang akan terjadi. Di dalam diagram blok sistem akan selalu ada komponen-komponen pokok seperti *set point*, elemen kontroler (*control unit*), elemen proses, elemen pengukuran (*sensing element* dan *transmitter*), *final control element*, dan sinyal keluaran. Dalam bentuk matematis semua elemen tersebut akan diisi oleh persamaan-persamaan matematis yang akan merepresentasikan karakteristik sistem dalam bentuk fungsi alih.



Gambar 2. 1. Diagram blok sistem kontrol

Pada sistem kontrol proses ada tiga parameter yang akan diukur yaitu *measured value* atau *process value* (nilai yang diukur sensor atau nilai keluaran dari proses), *manipulated value* (sinyal kontrol yang dikeluarkan oleh kontroler), dan *set value* (nilai yang diinginkan). Perubahan ketiga nilai ini akan mempengaruhi baik buruknya performansi dari sistem kontrol proses.

Dalam melihat performansi dari sistem kontrol digunakan analisis respon baik itu pendekatan orde 1 maupun orde 2. Pada Gambar 2. 2 terlihat respon orde 1 untuk analisis performansi dari suatu sistem kontrol proses. Dalam melihat performa sistem dilakukan analisis pada bagian transien dari respon. Sementara pada Gambar 2. 3 terlihat karakteristik dari respon orde 2, dimana K merupakan *gain overall* dari sistem.

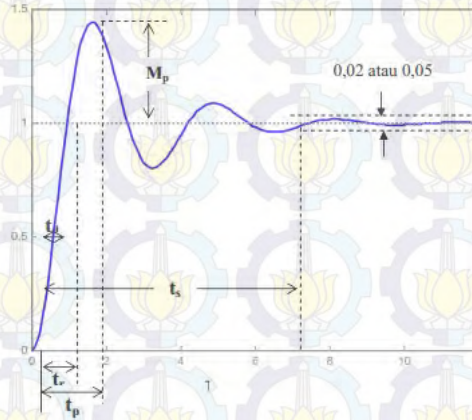


Gambar 2. 2. Spesifikasi respon orde satu

Karakteristik dari respon transien dengan masukan sinyal uji *unity step* adalah sebagai berikut [10]:

1. Konstanta waktu (τ), waktu yang dibutuhkan respon untuk mencapai kondisi 63.2% dari nilai keadaan tunak. Parameter ini diukur untuk melihat kecepatan respon.
2. Waktu tunda (t_d), yaitu waktu yang dibutuhkan respon untuk mencapai setengah dari nilai akhir yang ingin dicapai.
3. Waktu naik (t_r), yaitu waktu yang dibutuhkan respon dari keadaan 10% sampai 90%, 5% sampai 95%, atau 0% sampai 100% dari nilai akhir.
4. Waktu tunak (t_s), yaitu waktu yang dibutuhkan respon untuk mencapai keadaan untuk mencapai keadaan 2% atau 5% dari nilai akhir.
5. Waktu puncak (t_p), yaitu waktu yang dibutuhkan respon untuk mencapai puncak pertama *overshoot*.

6. Maksimum *overshoot* (M_p), yaitu nilai maksimum dari kurva respon diukur dari nilai satuan.



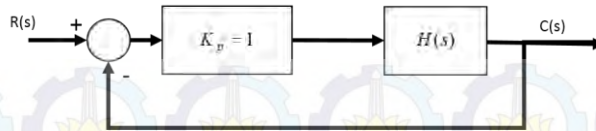
Gambar 2. 3. Spesifikasi respon orde 2

2.2. Identifikasi Sistem

Identifikasi sistem merupakan langkah yang dilakukan untuk mendapatkan model matematis dari *plant* berdasarkan hasil data pengukuran. Pengujian ini dilakukan dengan memberikan sinyal uji pada masukan. Model matematis yang didapat akan merepresentasikan hubungan antara masukan dan keluaran *plant*. Identifikasi sistem dapat dilakukan secara umpan balik terbuka maupun tertutup. Pada penelitian ini dilakukan identifikasi sistem dengan skema umpan balik tertutup. Untuk mempermudah perhitungan digunakan konstanta proporsional sebesar satu.

2.2.1. Identifikasi Umpan Balik Tertutup

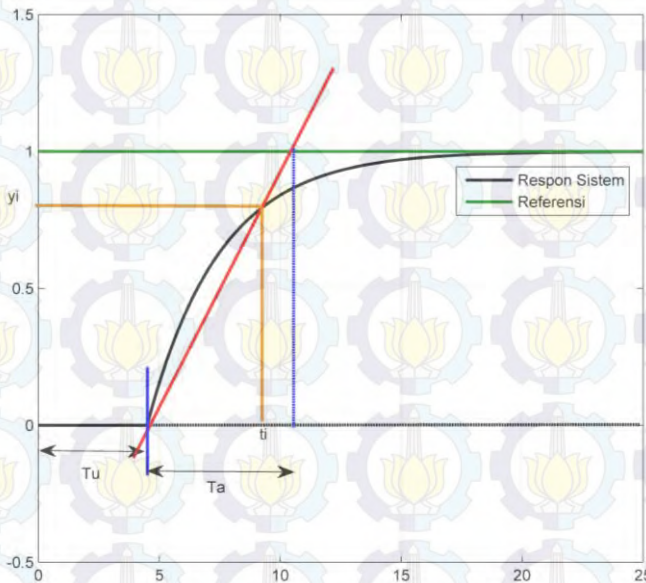
Identifikasi ini dilakukan dengan menambahkan kontroler proporsional (P) pada diagram blok sistem seperti terlihat pada Gambar 2. 4. Pada Gambar terlihat bahwa hasil dari identifikasi ini akan didapatkan fungsi alih dari model umpan balik tertutup *plant*. Kontroler P akan digunakan untuk mencari fungsi alih umpan balik terbuka *plant*.



Gambar 2. 4. Diagram blok umpan balik terbuka

2.2.2. Identifikasi Metode Strejc

Identifikasi dengan metode ini digunakan untuk respon berbentuk kurva S dengan waktu tunda seperti terlihat pada Gambar 2. 5. Untuk menentukan fungsi alih *plant*, maka perlu dilakukan beberapa tahap perhitungan parameter metode Strejc. Pada Tabel 2. 1 terlihat parameter yang menunjukkan perbedaan orde *plant*. Perbandingan nilai T_u dan T_a dari pada Gambar 2. 5 terlihat pada Tabel 2. 1.



Gambar 2. 5. Kurva S metode strejc

Tabel 2. 1. Parameter Metode Strejc

n	T_u/T	T_u/T	T_u/T_a
1	1.000	0.000	0.000
2	2.718	0.282	0.104
3	3.695	0.805	0.218
4	4.463	1.425	0.319
5	5.119	2.100	0.410

Mengacu pada respon yang ditunjukkan pada Gambar 2.5, maka dapat dipilih fungsi alih *plant* sesuai dengan Persamaan (2-1).

$$G(s) = \frac{K}{(1 + \tau s)^2} e^{-t_d s} \quad (2-1)$$

Untuk mencari parameter dari fungsi alih tersebut, maka dicari berdasarkan prosedur metode Strejc.

- Hitung nilai K, dimana K merupakan *gain overall*.
- Tarik garis yang memotong kurva S.
- Dapatkan nilai T_u dan T_a .
- Dapatkan nilai n, yaitu nilai n terbesar pada Tabel 2. 1 ketika nilai $\frac{T_u}{T_a}$ lebih besar dari $\frac{T_u}{T_a}$ pada tabel.
- Dapatkan nilai T berdasarkan nilai T_u/T pada Tabel 2. 1.
- Dapatkan nilai T_u' , dengan $T_u' = \left(\frac{T_u}{T_a} \right)_{tabel} T_a$.
- Hitung nilai $\tau = T_u - T_u'$.
- Jika nilai τ yang didapat bernilai negatif, maka nilai n yang digunakan salah. Pengerjaan diulangi ke langkah d.
- Masukkan parameter yang didapat pada Persamaan (2-1).

Dari fungsi alih umpan balik tertutup yang didapat, maka akan dihitung fungsi alih umpan balik terbuka sesuai dengan diagram blok pada Gambar 2. 4. Diagram blok ini mempunyai fungsi alih seperti pada Persamaan (2-2). Persamaan tersebut menunjukkan hubungan antara $G(s)$ dan $H(s)$.

$$G(s) = \frac{K_p H(s)}{1 + K_p H(s)} \quad (2-2)$$

Jika dipilih model matematis dari $H(s)$ adalah orde dua dengan waktu tunda, maka persamaan $H(s)$ dapat dituliskan sesuai dengan Persamaan (2-3). Waktu tunda yang terjadi dilakukan pendekatan dengan menggunakan deret Taylor.

$$H(s) = \frac{K e^{-\theta s}}{\alpha s^2 + \beta s + 1} \cong \frac{K(1 - \theta s)}{\alpha s^2 + \beta s + 1} \quad (2-3)$$

Dengan melakukan substitusi Persamaan (2-3) pada Persamaan (2-2), maka didapat persamaan $G(s)$ baru seperti Persamaan (2-8).

$$G(s) = \frac{K_p \frac{K(1 - \theta s)}{\alpha s^2 + \beta s + 1}}{1 + K_p \frac{K(1 - \theta s)}{\alpha s^2 + \beta s + 1}} \quad (2-4)$$

$$G(s) = \frac{\frac{K_p K(1 - \theta s)}{\alpha s^2 + \beta s + 1}}{\frac{\alpha s^2 + \beta s + 1 + K_p K(1 - \theta s)}{\alpha s^2 + \beta s + 1}} \quad (2-5)$$

$$G(s) = \frac{K_p K(1 - \theta s)}{\alpha s^2 + (\beta - K_p K \theta)s + 1 + K_p K} \quad (2-6)$$

$$G(s) = \frac{\frac{K_p K}{1 + K_p K} (1 - \theta s)}{\left(\frac{\alpha}{1 + K_p K} \right) s^2 + \left(\frac{\beta - K_p K \theta}{1 + K_p K} \right) s + 1} \quad (2-7)$$

$$G(s) = \frac{K'(1-\theta s)}{\alpha' s^2 + \beta' s + 1} \quad (2-8)$$

Dimana:

$$K' = \frac{K_p K}{1 + K_p K} \Leftrightarrow K = \frac{K'}{K_p (1 - K')} \quad (2-9)$$

$$\alpha' = \left(\frac{\alpha}{1 + K_p K} \right) \Leftrightarrow \alpha = \alpha' (1 + K_p K) \quad (2-10)$$

$$\beta' = \left(\frac{\beta - K_p K \theta}{1 + K_p K} \right) \Leftrightarrow \beta = \beta' (1 + K_p K) + K_p K \theta \quad (2-11)$$

2.3. Fungsi Alih dengan Waktu Tunda

Dalam sistem pengaturan proses, fenomena waktu tunda merupakan sesuatu yang tidak dapat dihindarkan. Waktu tunda ini dapat terjadi akibat pengukuran, aksi kontroler, dan operasi aktuator. Sistem yang memiliki waktu tunda dapat dimodelkan dengan menggunakan persamaan orde dua maupun orde satu yang memiliki waktu tunda. Persamaan ini disebut *second order plus time delay* (SOPDT) untuk orde dua dan *first order plus time delay* (FOPDT) untuk orde satu[9] dan dapat dituliskan sesuai Persamaan (2-12) dan (2-13).

$$G(s) = \frac{K}{\tau^2 s^2 + 2\zeta\tau s + 1} e^{-t_d s} \quad (2-12)$$

$$G(s) = \frac{K}{1 + \tau s} e^{-t_d s} \quad (2-13)$$

Dimana K merupakan penguatan proses, ζ merupakan koefisien redaman, τ merupakan konstan waktu, dan t_d adalah waktu tunda.

2.4. Process Control Technology (PCT-100)[12]

PCT-100 merupakan salah satu produk dari **Bytronic** yang dipergunakan untuk memperkenalkan serta mendemonstrasikan prinsip-prinsip dalam proses instrumentasi dan sistem kontrol proses. Pada Gambar 2. 6 terlihat bentuk dari PCT-100.

Peralatan ini menunjukkan proses yang terjadi pada tangki utama, dimana ada empat variabel proses yang diatur yaitu tekanan (*pressure*), level air, aliran air (*flow*), dan suhu (*temperature*). Aktuator yang terdapat pada PCT -100 adalah pompa dan pemanas. Pompa digunakan untuk menyalurkan air dari tangki penyimpanan menuju tangki utama. Sementara pemanas digunakan dalam proses sistem pengaturan suhu. Keempat variable tersebut diatur dengan menggunakan kontroler PID pada program bawaan alat ini [12].

PCT-100 juga dilengkapi dengan control module yang membuat pengguna dapat memeriksa seluruh komponen pada sistem jika terjadi kesalahan. PCT-100 dapat disambungkan dengan *Programmable Logic Controller* (PLC) dengan menggunakan *port* yang telah ada pada peralatan. PCT-100 dirancang khusus untuk skala laboratorium, agar para mahasiswa dapat melihat secara langsung sistem pengaturan proses yang ada di industri melalui simulator ini. Mahasiswa mampu menerapkan berbagai macam metode kontroler pada sistem pengaturan proses.



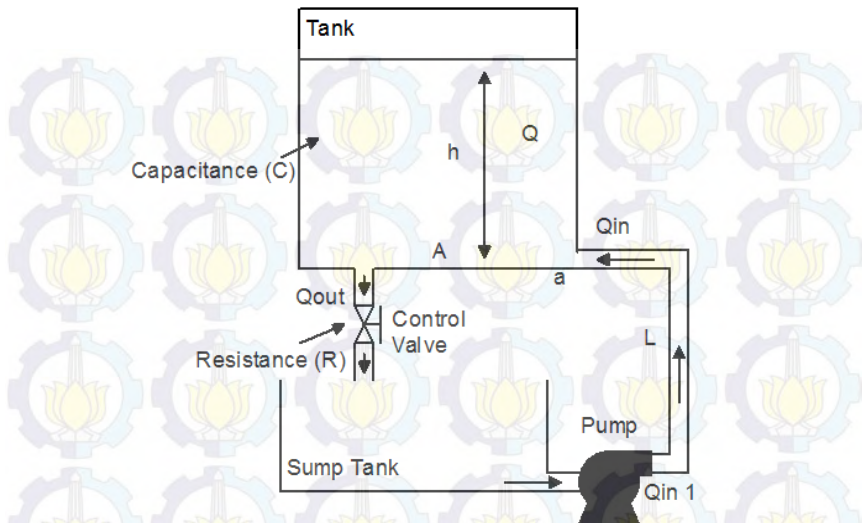
Gambar 2. 6. Process control technology[12]

2.5. Pemodelan Sistem

Sistem yang akan dirancang merupakan sistem pengaturan level pada tangki. Komponen dari sistem ini adalah *sump tank*, pompa, *flow sensor*, level sensor, tangki utama, dan *drain valve*. Representasi sistem tangki terlihat pada Gambar 2. 7. Aliran air akan dialirkan dari *sump tank* menuju ke tangki utama melalui pompa. Besarnya aliran air akan diukur oleh *flow sensor*. Pada tangki utama terdapat level sensor yang digunakan untuk mengukur ketinggian air pada tangki. *Drain valve* digunakan untuk mengembalikan air dari tangki utama menuju ke *sump tank* secara otomatis. Pada Tabel 2. 2 diterangkan variabel-variabel yang diamati dalam sistem pengaturan level PCT-100. Parameter ini akan digunakan untuk menurunkan persamaan matematis dari *plant* yang digunakan.

Tabel 2. 2. Parameter sistem

Simbol	Parameter	Satuan
Q	Laju penyimpanan air dalam tangki	cm^3 / s
Q_{in}	Debit air yang masuk ke tangki	cm^3 / s
Q_{in1}	Debit air yang dikeluarkan pompa	cm^3 / s
Q_{out}	Debit air yang keluar dari tangki	cm^3 / s
A	Luas penampang tangki	cm^2
a	Luas penampang pipa	cm^2
C_d	Konstanta kapasitas	-
h	Ketinggian air	cm
h_o	Ketinggian air tangki (titik kerja)	cm
Q_0	Debit air yang keluar dari tangki (titik kerja)	cm^3 / s
V_p	Volume pipa dari pompa menuju tangki	cm^2
L	Panjang pipa	cm
R	Resistansi	s / cm^2
C	Capasitansi	m^2



Gambar 2. 7. Representasi sistem tanki

2.5.1. Perhitungan Waktu tunda

Pompa dan tanki pada PCT-100 dihubungkan dengan pipa yang memiliki panjang 90 cm dan diameter 1 cm. Maka akan dimodelkan waktu tunda pada pemodelan ini, dimana waktu tunda yang dihitung sesuai dengan persamaan (2-14).

$$L = \frac{V_p}{Q} \quad (2-14)$$

Dimana V merupakan volume dari pipa dan Q adalah aliran air yang mengalir pada pipa. Dengan adanya waktu tunda maka aliran air yang masuk ke tanki akan berbeda dengan aliran yang dikeluarkan pompa. Jika aliran air yang masuk ke dalam tanki disimbolkan Q_{in} dan aliran air yang dikeluarkan pompa disimbolkan Q_{in1} . Maka dapat dirumuskan seperti pada Persamaan (2-15).

$$Q_{in}(t) = Q_{inl}(t + L) \quad (2-15)$$

Dengan melakukan transformasi Laplace pada Persamaan (2-15), maka didapat Persamaan (2-16) yang menyatakan hubungan antara debit air yang dikeluarkan pompa dan debit air yang masuk kedalam tanki.

$$\frac{Q_{in}(s)}{Q_{inl}(s)} = e^{-Ls} \quad (2-16)$$

2.5.2. Pemodelan Tanki

Pada pemodelan tanki dikenal istilah kapasitansi dan resistansi. Kapasitansi dari tanki didefinisikan sebagai perubahan volume air didalam tanki yang menyebabkan perubahan ketinggian air. Rumus dari kapasitansi sesuai dengan Persamaan (2-17). Kapasitansi tanki sama dengan luar penampang tanki (A).

$$C = A = \frac{\text{Perubahan Volume Air (cm}^3\text{)}}{\text{Perubahan Ketinggian Air (cm)}} = \frac{1}{4} \pi d^2 \quad (2-17)$$

Sementara resistansi dari tanki didefinisikan sebagai perubahan ketinggian air didalam tanki yang menyebabkan perubahan debit air. Rumus dari resistansi sesuai dengan Persamaan (2-18). Nilai R akan bergantung pada jenis aliran air yang terdapat pada pipa keluar. Pada sistem yang digunakan pada penelitian ini, aliran air berjenis *laminar* yang berarti aliran air bersifat linear. Hukum pada aliran air *laminar* dapat dianalogikan seperti Hukum Coulomb's, dimana arus yang mengalir berbanding lurus dengan beda potensial. Maka resistansi pada sistem ini dapat ditulis seperti Persamaan (2-19).

$$R = \frac{\text{Perubahan Ketinggian Air (cm)}}{\text{Perubahan Debit Air (cm)}} \quad (2-18)$$

$$R = \frac{dH}{dQ} = \frac{H}{Q} \quad (2-19)$$

Dalam menurunkan persamaan matematis maka ditentukan masukan sistem yaitu aliran masuk Q_{in} dan keluaran berupa ketinggian tangki (h). Untuk persamaan kesetimbangan pada tangki terlihat pada Persamaan (2-20).

$$Q_{in} = Q + Q_{out} \quad (2-20)$$

Jika A adalah luas penampang tangki dan h adalah ketinggian air pada tangki, maka Persamaan (2-21) menunjukkan kaitan antara A dan h .

$$Q_{in} = A \frac{dh}{dt} + Q_{out} \quad (2-21)$$

Dengan menentukan h_0 dan Q_0 sebagai titik kerja, maka Persamaan (2-21) dapat ditulis seperti Persamaan (2-22). Pada Persamaan (2-23) didapatkan Q_0 sebagai hubungan h_0 dan R . Maka persamaan *differensial* dari sistem ini, dapat dituliskan seperti Persamaan (2-24) sebagai hubungan antara ketinggian dan debit air terhadap perubahan nilai masukan.

$$Q_{in} = A \frac{dh_0}{dt} + Q_0 \quad (2-22)$$

$$Q_{in} = A \frac{dh_0}{dt} + \frac{h_0}{R} \quad (2-23)$$

$$\frac{Q_{in}}{dt} = A \frac{dh_0}{dt} + \frac{h_0}{R} \quad (2-24)$$

Dari Persamaan (2-24), dengan melakukan transformasi Laplace didapatkan fungsi alih sistem seperti persamaan (2-25).

$$\frac{H(s)}{Q(s)} = \frac{R}{RA s + 1} \quad (2-25)$$

2.5.3. Pendekatan Deret Taylor

Pendekatan deret Taylor merupakan metode pendekatan dari pemodelan waktu tunda yang terjadi pada sistem. Pemodelan dari e^{-sT} terlihat pada Persamaan (2-26).

$$e^{-\theta s} \cong 1 - \theta s + \frac{\theta^2 s^2}{2!} - \frac{\theta^3 s^3}{3!} + \frac{\theta^4 s^4}{4!} \dots \quad (2-26)$$

Jika T yang terjadi sangat kecil maka pendekatan dari $e^{-\theta s}$ dapat dipilih orde pertama seperti pada Persamaan (2-27).

$$e^{-\theta s} \cong 1 - \theta s \quad (2-27)$$

2.6. Kontroler PID

Kontroler PID merupakan kontroler *feed-forward* yang berfungsi mengolah sinyal kesalahan menjadi sinyal kontrol, dimana hubungan sinyal kontrol terhadap sinyal kesalahan dapat berupa proporsional, integral, differensial, ataupun gabungan diantaranya. Struktur kontroler PID terlihat pada Persamaan (2-28).

$$\frac{U(s)}{E(s)} = \left(K_p + \frac{K_i}{s} + K_d s \right) \quad (2-28)$$

Karakteristik dari tiga jenis kontroler ini adalah kontroler proporsional (K_p) mempunyai kelebihan dalam mengurangi waktu naik dan akan memperkecil *error steady state* (e_{ss}) tapi tidak menghilangkannya. Kontroler integral (K_i) memiliki efek untuk menghilangkan e_{ss} terhadap masukan konstan maupun *step*, tapi hal tersebut akan membuat respon menjadi lebih lama. Sementara kontroler *derivative* (K_d) mempunyai efek dalam meningkatkan kestabilan dari sistem, mengurangi overshoot, dan meningkatkan performa respon transient [14]. Efek dari ketiga parameter kontroler ini dalam sistem kontrol umpan balik tertutup terlihat pada Tabel 2. 3.

Tabel 2. 3. Perbandingan parameter k_p , k_i , dan k_d

Respon umpan balik tertutup	t_r	Overshoot	t_s	e_{ss}
K_p	Turun	Naik	Perubahan Kecil	Turun
K_i	Turun	Naik	Naik	Menghilangkan
K_d	Perubahan Kecil	Turun	Turun	Tidak ada perubahan

Berdasarkan bentuk hubungan sinyal kesalahan dan sinyal kontrol, kontroler ini dibagi atas beberapa tipe yaitu kontroler P, kontroler PI, kontroler PD, dan kontroler PID.

2.6.1. Kontroler P (*Proportional Controller*)

Hubungan sinyal kesalahan dan sinyal kontrol pada kontroler tipe-P dapat dinyatakan dalam persamaan (2-29). Dengan melakukan transformasi Laplace didapatkan fungsi alih seperti pada Persamaan (2-30)

$$u(t) = K_p \cdot e(t) \quad (2- 29)$$

$$\frac{U(s)}{E(s)} = K_p \quad (2- 30)$$

2.6.2. Kontroler PI (*Proportional + Integral Controller*)

Hubungan sinyal kesalahan dan sinyal kontrol pada kontroler tipe-PI dapat dinyatakan dalam persamaan (2-31). Dengan melakukan transformasi Laplace didapatkan fungsi alih seperti pada Persamaan (2-32)

$$u(t) = K_p [e(t) + \frac{1}{\tau_i} \int e(t) dt] \quad (2- 31)$$

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_i s} \right) \quad (2- 32)$$

2.6.3. Kontroler PD (*Proportional + Derivative Controller*)

Hubungan sinyal kesalahan dan sinyal kontrol pada kontroler tipe-PD dapat dinyatakan dalam persamaan (2-33). Dengan melakukan transformasi Laplace didapatkan fungsi alih seperti pada Persamaan (2-34)

$$u(t) = K_p [e(t) + \tau_d \frac{d}{dt} e(t)] \quad (2- 33)$$

$$\frac{U(s)}{E(s)} = K_p (1 + \tau_d s) \quad (2- 34)$$

2.6.4. Kontroler PID (*Proportional + Integral + Derivative Controller*)

Pada kontroler PID ini terbagi dua yaitu kontroler PID standar dan PID modifikasi.

a. Kontroler PID Standar

Hubungan sinyal kesalahan dan sinyal kontrol pada kontroler tipe-PID standar dapat dinyatakan dalam persamaan (2-35). Dengan melakukan transformasi Laplace didapatkan fungsi alih seperti pada Persamaan (2-36)

$$u(t) = K_p [e(t) + \frac{1}{\tau_i} \int e(t) dt + \tau_d \frac{d}{dt} e(t)] \quad (2- 35)$$

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \quad (2- 36)$$

b. Kontroler PID Modifikasi

Hubungan sinyal kesalahan dan sinyal kontrol pada kontroler tipe-PID modifikasi dapat dinyatakan dalam fungsi alih pada persamaan (2-37).

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_i s} + \frac{\tau_d s}{\tau_d s + 1} \right) \quad (2-37)$$

2.7. Model Predictive Control (MPC)

MPC pertama kali dikenalkan oleh Richalet, 1978 yaitu MAC (*Model Algorithmic Control*) dan Cutler dan Ramaker, 1979 yaitu DMC (*Dynamix Matric Control*). MPC dirancang berdasarkan model matematika dari *plant*. Model yang digunakan dalam sistem kontrol ini dalam bentuk *state space*. Dengan menggunakan bentuk *state space*, maka informasi yang dibutuhkan untuk memprediksi nilai kedepan direpresentasikan ke dalam *variabel state* pada waktu tersebut.

2.7.1. Penurunan Persamaan Kontroler Prediktif

a. Mendapatkan Matrik *Augmented*

Dalam perancangan kontroler prediktif [15], pemodelan sistem dilakukan dalam bentuk *state space* dimana jika sistem *single input* dan *single output* akan memiliki persamaan seperti yang ditunjukkan pada Persamaan (2-38) dan (2-39).

$$x_m(k+1) = A_m x_m(k) + B_m u(k) \quad (2-38)$$

$$y(k) = C_m x_m(k) \quad (2-39)$$

$u(k)$ merupakan sinyal masukan dan y adalah keluaran proses serta x_m merupakan matrik *state* dari sistem. Dengan melakukan pergeseran sebanyak k pada persamaan (2-38) maka akan menjadi Persamaan (2-40).

$$x_m(k+1) - x_m(k) = A_m (x_m(k) - x_m(k-1)) + B_m (u(k) - u(k-1)) \quad (2-40)$$

Misalkan perubahan nilai pada *variabel state* dan nilai keluaran seperti pada persamaan (2-41), (2-42) dan (2-43).

$$\Delta x_m(k+1) = x_m(k+1) - x_m(k) \quad (2-41)$$

$$\Delta x_m(k) = x_m(k) - x_m(k-1) \quad (2-42)$$

$$\Delta u(k) = u(k) - u(k-1) \quad (2-43)$$

Substitusikan persamaan (2-41), (2-42) dan (2-43) ke dalam persamaan (2-40) menjadi Persamaan (2-44).

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) \quad (2-44)$$

Hubungan antara variabel *state* dengan nilai keluaran dimasukkan kedalam variabel baru seperti pada Persamaan (2-45).

$$x(k) = [\Delta x_m(k)^T \ y(k)]^T \quad (2-45)$$

Pada Persamaan (2-46) terlihat persamaan *ouput* dengan variabel baru hasil substitusi dari persamaan (2-42)

$$\begin{aligned} y(k+1) - y(k) &= C_m(x_m(k+1) - x_m(k)) = C_m \Delta x_m(k+1) \\ &= C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) \end{aligned} \quad (2-46)$$

Dengan memasukkan persamaan (2-44) dan (2-46) ke dalam model *state space* seperti pada Persamaan (2-47).

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) \quad (2-47)$$

$$y(k) = \begin{bmatrix} o_m & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \quad (2-48)$$

Dimana nilai $o_m = [0 \ 0 \ \dots \ 0]$. Maka didapat *matrik augmented* seperti pada persamaan (2-49), (2-50), dan (2-51).

$$A_{-e} = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & 1 \end{bmatrix} \quad (2-49)$$

$$B_{-e} = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \quad (2-50)$$

$$C_{-e} = \begin{bmatrix} o_m & 1 \end{bmatrix} \quad (2-51)$$

b. Prediksi Variable State dan Keluaran

Setelah melakukan penurunan terhadap model matematis *plant*, maka langkah selanjutnya dalam merancang kontroler prediktif adalah melakukan perhitungan terhadap nilai keluaran yang diprediksi dengan sinyal kontrol yang berubah. Asumsikan *sampling* dilakukan tiap satuan k_i dan $x(k_i)$ merupakan vektor *state*. $x(k_i)$ bertujuan memberikan informasi terkini dari kondisi *plant*. Pada Persamaan (2-52) terlihat notasi dari sinyal kontrol.

$$\Delta u(k_i), \Delta u(k_i + 1), \dots, \Delta u(k_i + N_c - 1) \quad (2- 52)$$

Dimana N_c adalah *control horizon*. Pada Persamaan (2-53) terlihat notasi dari variabel *state*. Dimana N_p adalah *prediction horizon*.

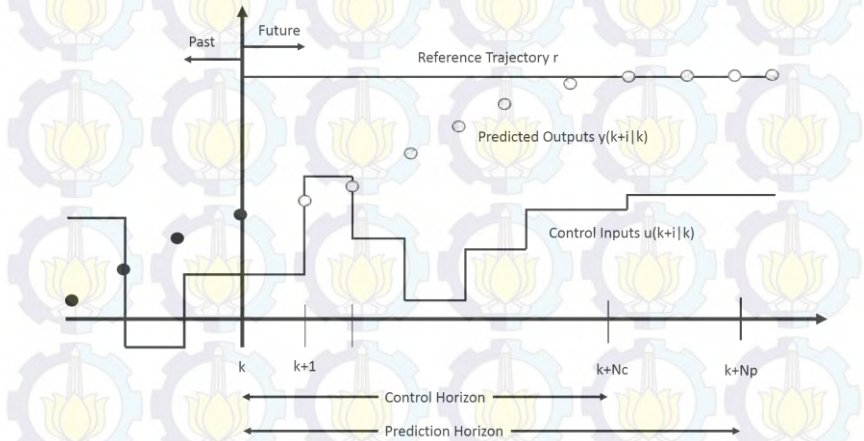
$$x(k_i + 1 | k_i), x(k_i + 2 | k_i), \dots, x(k_i + m | k_i), \dots, x(k_i + N_p | k_i), \quad (2- 53)$$

Diketahui bahwa $x(k_i + 1 | k_i)$ berarti bahwa sistem memprediksi variabel state $k_i + 1$ dengan diberikan informasi *plant* saat $x(k_i)$. Metode *Model Predictive Control* terlihat sesuai Gambar 2. 8. Dalam metode ini sinyal keluaran diprediksi sesuai perhitungan sinyal kontrol yang berdasarkan objektif kontrol yaitu meminimumkan kesalahan. Sinyal yang diprediksi sebanyak N_p . Berdasarkan matrik *augmented*, maka didapatkan prediksi dari variabel state seperti pada Persamaan (2-54).

$$\begin{aligned} x(k_i + 1 | k_i) &= Ax(k_i) + B\Delta u(k_i) \\ x(k_i + 2 | k_i) &= Ax(k_i + 1 | k_i) + B\Delta u(k_i + 1) \\ &= A^2 x(k_i) + AB\Delta u(k_i) + B\Delta u(k_i + 1) \\ x(k_i + N_p | k_i) &= A^{N_p} x(k_i) + A^{N_p - 1} B\Delta u(k_i) + A^{N_p - 2} B\Delta u(k_i + 1) \\ &= + \dots + A^{N_p - N_c} B\Delta u(k_i + N_c - 1) \end{aligned} \quad (2- 54)$$

Dari Persamaan (2-51) dapat diturunkan persamaan keluaran sistem yaitu seperti pada Persamaan (2-55).

$$\begin{aligned}
y(k_i + 1 | k_i) &= CAx(k_i) + CB\Delta u(k_i) \\
y(k_i + 2 | k_i) &= CAx(k_i + 1 | k_i) + CB\Delta u(k_i + 1) \\
&= CA^2x(k_i) + CAB\Delta u(k_i) + CB\Delta u(k_i + 1) \\
y(k_i + N_p | k_i) &= CA^{N_p}x(k_i) + CA^{N_p-1}B\Delta u(k_i) \\
&= + \dots + CA^{N_p-N_c}B\Delta u(k_i + N_c - 1)
\end{aligned} \tag{2-55}$$



Gambar 2. 8. Metode model predictive control

Dengan memasukkan Persamaan (2-55) pada persamaan keluaran dan state maka didapat vektor keluaran dan sinyal kontrol seperti pada Persamaan (2-56) dan (2-57).

$$Y = [y(k_i + 1 | k_i) y(k_i + 2 | k_i) y(k_i + 3 | k_i) \dots y(k_i + N_p | k_i)]^T \tag{2-56}$$

$$\Delta U = [\Delta u(k_i) \Delta u(k_i + 1) \Delta u(k_i + 2) \dots \Delta u(k_i + N_c - 1)]^T \tag{2-57}$$

Dengan menggabungkan Persamaan (2-56) dan (2-57) maka didapat persamaan matrik keluaran seperti pada Persamaan (2-58).

$$Y = Fx(k_i) + \Phi \Delta u \tag{2-58}$$

Dimana:

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \dots \\ CA^{N_p} \end{bmatrix}$$

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix}$$

c. Optimisasi

Dengan nilai *set point* sebesar $r(k_i)$ pada waktu *sampling* sebesar k_i , maka objektif dari kontroler adalah membawa nilai keluaran hasil prediksi sedekat mungkin dengan nilai *set point*. Jika suatu masukan memiliki nilai konstan pada jendela *horizon* maka objektif kontrol berubah menjadi perancangan sinyal kontrol terbaik yang dapat meminimalisasi kesalahan antara *set point* dengan keluaran hasil prediksi. Asumsikan vektor R yang berisi informasi *set point* seperti terlihat pada Persamaan (2-59). Dan pada Persamaan (2-60) terlihat *cost function* atau objektif kontrol yang dilambangkan huruf J .

$$R_s^T = [1 \ 1 \ \dots \ 1]r(k_i) \quad (2-59)$$

$$J = (R_s - Y)^T (R_s - Y) + \Delta u^T R \Delta u \quad (2-60)$$

Dimana elemen pertama dari J berisikan objektif dalam meminimumkan kesalahan antara sinyal keluaran dan *set point*. Sementara pada elemen kedua berisikan ukuran dari sinyal kontrol. R merupakan matrik diagonal yang didapatkan dari $R = r_w I_{N_c \times N_c}$ dan r_w

merupakan parameter tuning dimana nilainya $r_w \geq 0$. Untuk mencari Δu yang meminimalkan fungsi J, maka substitusikan Persamaan (2-58) kedalam Persamaan (2-60) menjadi Persamaan (2-61).

$$J = A + \Delta u^T (\phi^T \phi + R) \Delta u \quad (2- 61)$$

Dimana:

$$A = (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - 2\Delta u^T \phi^T (R_s - Fx(k_i))$$

Untuk meminimumkan fungsi J maka didapatkan Persamaan (2-62).

$$\frac{\partial J}{\partial \Delta u} = -2\phi^T (R_s - Fx(k_i)) + 2(\phi^T \phi + R)\Delta u \quad (2- 62)$$

$$\frac{\partial J}{\partial \Delta u} = 0 \quad (2- 63)$$

Sesuai dengan Persamaan (2-63) maka didapat solusi optimal dari sinyal kontrol terlihat pada Persamaan (2-64). Pada Gambar 2. 9 terlihat diagram blok *predictive control*.

$$\Delta U = (\Phi^T \Phi + R)^{-1} \Phi^T (R_s - Fx(k_i)) \quad (2- 64)$$

d. **Receding Horizon Control**

Meskipun vektor sinyal kontrol adalah $\Delta u(k_i)\Delta u(k_i + 1)\Delta u(k_i + 2).....\Delta u(k_i + N_c - 1)$, dengan menerapkan hukum *receding horizon control*, maka yang diambil hanya elemen pertama yaitu $\Delta u(k_i)$. Untuk kondisi berikutnya akan diambil *initial state* yang baru sesuai dengan pengukuran yang dilakukan.

2.7.2 Sistem Pengaturan Umpan Balik Tertutup

Dengan menerapkan hukum *receding horizon control* pada persamaan $\Delta u(k_i)$ maka didapat Persamaan (2-65).

$$\Delta u(k_i) = [1 \ 0 \ . \ . \ 0](\Phi^T \Phi + R)^{-1}(\Phi^T R_s r(k_i) - \Phi^T Fx(k_i)) \quad (2- 65)$$

$$\Delta u(k_i) = K_y r(k_i) - K_{mpc} x(k_i) \quad (2-66)$$

Dimana:

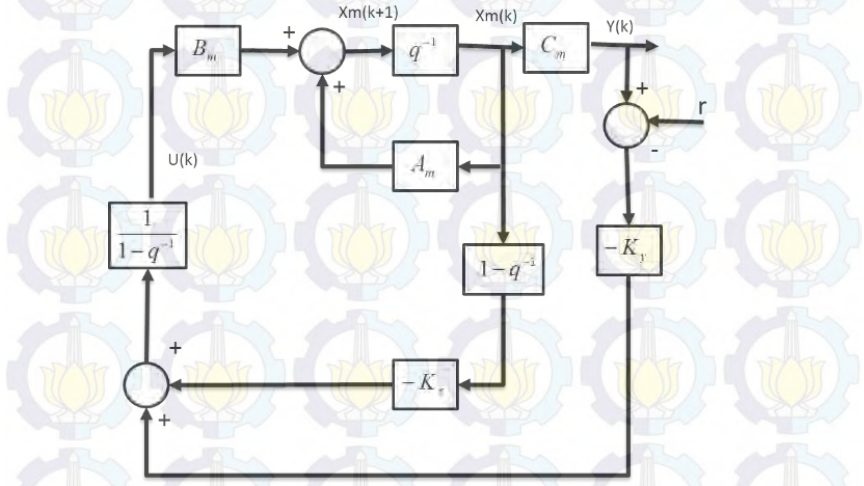
K_y adalah element pertama dari persamaan $(\Phi^T \Phi + R)^{-1} \Phi^T R_y$,

K_{mpc} adalah gain *state feedback* baris pertama dari $(\Phi^T \Phi + R)^{-1} \Phi^T F$

Dengan model matrik *augmented* seperti Persamaan (2-66), maka didapat persamaan umpan balik tertutup sistem pada Persamaan (2-68).

$$x(k+1) = Ax(k) + B\Delta u(k) \quad (2-67)$$

$$x(k+1) = (A - BK_{mpc})x(k) + BK_y r(k) \quad (2-68)$$



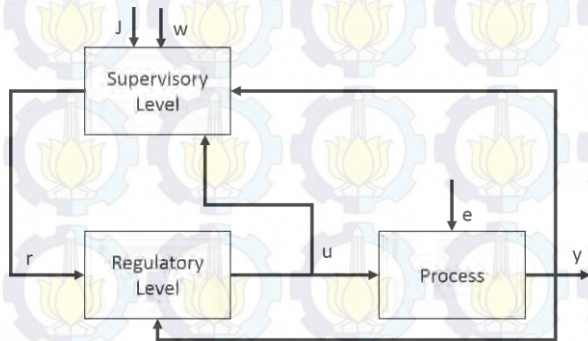
Gambar 2. 9. Blok diagram predictive control

Nilai gain K_y akan identik dengan nilai gain K_{mpc} pada kolom kedua. Maka dapat ditulis bahwa $K_{mpc} = [K_x \ K_y]$, karena vektor *state* $x(k) = [\Delta x_m(k)^T \ y(k)]^T$ maka dapat dituliskan bahwa K_x merupakan *gain state feedback* terhadap $\Delta x_m(k)$ dan K_y merupakan *gain state feedback* terhadap $y(k)$.

Pada Gambar 2. 9 menunjukkan struktur *state feedback* untuk *discrete model prediction control* (DMPC) dimana q^{-1} adalah operator *backward shift*. Dalam sistem ini yang bertindak sebagai integrator adalah $\frac{1}{1-q^{-1}}$. Am, Bm, dan Cm merupakan parameter persamaan *state space* dari sistem. K_x dan K_y merupakan *gain state feedback*.

2.8. Supervisory Predictive Control (SPC)

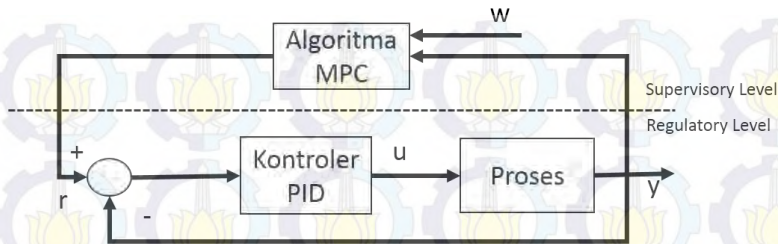
Pada konfigurasi sistem ini terdapat dua level yaitu *supervisory* dan *regulatory*. Level *supervisory* secara dinamis akan mengoptimalkan fungsi objektif dan akan memberikan *set point* pada level *regulatory*. Pada Gambar 2. 10 menunjukkan konfigurasi umum dari *supervisory predictive control*. Variabel lainnya dalam struktur ini adalah referensi eksternal (w), variabel kontrol (y), sinyal kontrol (u), dan gangguan tidak terukur (e) [1].



Gambar 2. 10. Konfigurasi SPC

2.8.1 Konfigurasi Cascade SPC

Pada konfigurasi *cascade*, MPC digunakan untuk merubah *set point* yang telah ada pada sistem kontrol umpan balik tertutup yang terdiri atas kontroler (C) dan *plant* (P) dengan *feedback*. Struktur dari konfigurasi *cascade* terlihat pada Gambar 2. 11[2]. Pada konfigurasi ini kedua kontroler yaitu pada level *supervisory* dan *regulatory* disusun secara seri. Bentuk persamaan dari *plant* dan kontroler yang digunakan dirumuskan dalam bentuk *state space*.



Gambar 2. 11. Konfigurasi *cascade* SPC

Persamaan *plant*:

$$x_p(k+1) = A_p x_p(k) + B_{p1} u(k) + B_{p2} w(k) \quad (2-69)$$

$$y(k) = C_p x_p(k) + D_{p1} u(k) + D_{p2} w(k) \quad (2-70)$$

Persamaan kontroler:

$$x_c(k+1) = A_c x_c(k) + B_c e(k) \quad (2-71)$$

$$u(k) = C_c x_c(k) + D_c e(k) + d_k \quad (2-72)$$

Dimana $w(k)$ merupakan gangguan pada *plant* dan d_k adalah *noise* pada sinyal kontrol. Dari konfigurasi ini dapat dirumuskan kesalahan pada keluaran seperti pada Persamaan (2-73).

$$e_k = r_k - y_k \quad (2-73)$$

Persamaan (2-69) dan (2-72) didapatkan dari penurunan model matematis *plant* menggunakan rumus dasar fisika atau melalui proses identifikasi. Persamaan kontroler didapat dari penurunan rumus dari kontroler yang telah digunakan. Objektif kontrol dari penerapan algoritma MPC adalah untuk mengatur keluaran *plant* agar sesuai dengan *set point*. Substitusi Persamaan (2-72) ke dalam Persamaan (2-73) dan didapat Persamaan (2-74).

$$e_k = r_k - y_k = r_k - C_p x_p(k) - D_{p1} u(k) - D_{p2} w(k) \quad (2-74)$$

Substitusikan Persamaan (2-74) ke dalam Persamaan (2-72) untuk mendapatkan nilai u_k terlihat pada Persamaan (2-75).

$$\begin{aligned} u(k) &= C_c x_c(k) + D_c(r_k - C_p x_p(k) - D_{p1} u(k) - D_{p2} w(k)) + d_k \\ &= M(C_c x_c(k) + D_c r_k - D_c C_p x_p(k) - D_c D_{p2} w(k) + d_k) \end{aligned} \quad (2-75)$$

Dimana $M = (I + D_c D_{p1})^{-1}$. Dengan melakukan substitusi Persamaan (2-74) dan (2-75) pada Persamaan (2-61), maka didapat Persamaan (2-76).

$$\begin{aligned} x_c(k+1) &= A_c x_c(k) + B_c r_k - B_c C_p x_p(k) - B_c D_{p1} M C_c x_c(k) \\ &\quad - B_c M D_{p1} D_c r_k + B_c M D_{p1} D_c C_p x_p(k) \\ &\quad - B_c M D_{p1} D_c D_{p2} w(k) + B_c M D_{p1} d_k - B_c D_{p2} w(k) \end{aligned} \quad (2-76)$$

Substitusikan u_k pada Persamaan (2-69) dan (2-70) dan didapatkan Persamaan (2-77) dan (2-78).

$$\begin{aligned} x_p(k+1) &= A_p x_p(k) + B_{p1} M C_c x_c(k) + B_{p1} M D_c r_k - B_{p1} M D_c C_p x_p(k) \\ &\quad - B_{p1} M D_c D_{p2} w(k) + B_{p1} M d_k + B_{p2} w(k) \end{aligned} \quad (2-77)$$

$$\begin{aligned} y(k) &= C_p x_p(k) + D_{p1} M C_c x_c(k) + D_{p1} M D_c r_k - D_{p1} M D_c C_p x_p(k) \\ &\quad - D_{p1} M D_c D_{p2} w(k) + D_{p1} M d_k + D_{p2} w(k) \end{aligned} \quad (2-78)$$

Persamaan (2-76), (2-77), dan (2-78) dapat dituliskan dalam bentuk matrik seperti Persamaan (2-79) dan (2-80).

$$X_{k+1} = A X_k + B_1 U_k + B_2 W_k \quad (2-79)$$

$$Y_k = C X_k + D_1 U_k + D_2 W_k \quad (2-80)$$

Dimana:

$$X_k = \begin{bmatrix} x_p(k) \\ x_c(k) \end{bmatrix}, \quad U_k = r_k, \quad W_k = \begin{bmatrix} w_k \\ d_k \end{bmatrix}, \quad Y_k = \begin{bmatrix} y_k \\ u_k \end{bmatrix} \quad (2-81)$$

$$A = \begin{bmatrix} A_p - B_{p1}MD_cC_p & B_{p1}MC_c \\ B_cD_{p1}MD_cC_p - B_cC_p & A_c - B_cD_{p1}MC_c \end{bmatrix} \quad (2- 82)$$

$$B_1 = \begin{bmatrix} B_{p1}MD_c \\ B_c - B_cD_{p1}MC_c \end{bmatrix} \quad (2- 83)$$

$$B_2 = \begin{bmatrix} B_{p2} - B_{p1}MD_cD_{p2} & B_{p1}M \\ B_cD_{p1}MD_cD_{p2} - B_cD_{p2} & -B_cD_{p1}M \end{bmatrix} \quad (2- 84)$$

$$C = \begin{bmatrix} C_p - D_pMD_cC_p & D_{p1}MC_c \\ -MD_cC_p & MC_c \end{bmatrix} \quad (2- 85)$$

$$D_1 = \begin{bmatrix} D_{p1}MD_c \\ MD_c \end{bmatrix} \quad (2- 86)$$

$$D_2 = \begin{bmatrix} D_{p2} - D_{p1}MD_cD_{p2} & D_{p1}M \\ -MD_cD_{p2} & M \end{bmatrix} \quad (2- 87)$$

2.8.2 Konfigurasi *Parallel SPC*

Pada konfigurasi *parallel*, MPC digunakan secara *parallel* dengan kontroler PI yang telah ada. Hal ini bertujuan untuk meningkatkan performansi sistem dari sistem pengaturan umpan balik tertutup yang telah ada. Sama dengan konfigurasi *cascade*, konfigurasi ini juga mengasumsikan bahwa tidak ada perubahan struktur dari sistem umpan balik tertutup yang telah ada. Struktur dari konfigurasi *parallel* terlihat pada Gambar 2. 12[2] yang memiliki dua sinyal kontrol yaitu berasal dari kontoler PI dan MPC. Bentuk persamaan dari *plant* dan kontroler yang digunakan dirumuskan dalam bentuk *state space*.

Persamaan *plant*:

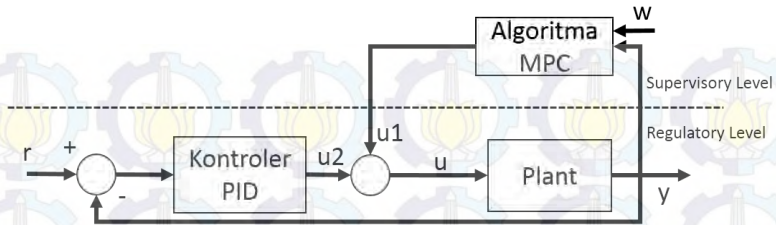
$$x_p(k+1) = A_px_p(k) + B_{p1}u(k) + B_{p2}w(k) \quad (2- 88)$$

$$y(k) = C_px_p(k) + D_{p1}u(k) + D_{p2}w(k) \quad (2- 89)$$

Persamaan kontroler:

$$x_c(k+1) = A_cx_c(k) + B_ce(k) \quad (2- 90)$$

$$u_2(k) = C_cx_c(k) + D_ce(k) + d_k \quad (2- 91)$$



Gambar 2. 12. Konfigurasi *parallel* SPC

Dimana kesalahan pada keluaran dan masukan *plant* dirumuskan seperti pada Persamaan (2-92).

$$e_k = r_k - y_k, \quad u_k = u_{k1} - u_{k2} \quad (2-92)$$

Dengan melakukan penurunan rumus yang sama seperti pada konfigurasi *cascade*, maka didapatkan persamaan matrik seperti pada Persamaan (2-93) dan (2-94).

$$X_{k+1} = AX_k + B_1U_k + B_2W_k \quad (2-93)$$

$$Y_k = CX_k + D_1U_k + D_2W_k \quad (2-94)$$

Dimana:

$$X_k = \begin{bmatrix} x_p(k) \\ x_c(k) \end{bmatrix}, \quad U_k = \begin{bmatrix} r_k \\ u_{k1} \end{bmatrix}, \quad W_k = \begin{bmatrix} w_k \\ d_k \end{bmatrix}, \quad Y_k = \begin{bmatrix} y_k \\ u_k \end{bmatrix} \quad (2-95)$$

$$A = \begin{bmatrix} A_p - B_{p1}MD_cC_p & B_{p1}MC_c \\ B_cD_{p1}MD_cC_p - B_cC_p & A_c - B_cD_{p1}MC_c \end{bmatrix} \quad (2-96)$$

$$B_1 = \begin{bmatrix} B_{p1}MD_c & B_{p1} - B_{p1}MD_cD_{p1} \\ B_c - B_cD_{p1}MD_c & B_cD_{p1}MD_cD_{p1} - B_cD_{p1} \end{bmatrix} \quad (2-97)$$

$$B_2 = \begin{bmatrix} B_{p2} - B_{p1}MD_cD_{p2} & B_{p1}M \\ B_cD_{p1}MD_cD_{p2} - B_cD_{p2} & -B_cD_{p1}M \end{bmatrix} \quad (2-98)$$

$$C = \begin{bmatrix} C_p - D_{p1}MD_cC_p & D_{p1}MC_c \\ -MD_cC_p & MC_c \end{bmatrix} \quad (2-99)$$

$$D_1 = \begin{bmatrix} D_{p1}MD_c D_{p1} - D_{p1}MD_c D_{p1} \\ MD_c \quad I - MD_c D_{p1} \end{bmatrix} \quad (2-100)$$

$$D_2 = \begin{bmatrix} D_{p2} - D_{p1}MD_c D_{p2} & D_{p1}M \\ -MD_c D_{p2} & M \end{bmatrix} \quad (2-101)$$

2.8.3 Penurunan Rumus SPC

a. Inisialisasi parameter

Dengan menggunakan fungsi Laguerre, maka didapat persamaan sinyal kontrol seperti pada Persamaan (2-102).

$$\Delta u(k_i + k) = \sum_{j=1}^N c_j(k_i) l_j(k) \quad (2-102)$$

Dapat dituliskan dalam fungsi *Laguerre*,

$$\Delta u(k_i + k) = L(k)^T \eta \quad (2-103)$$

$$\eta = [c_1 \quad c_2 \quad \dots \quad c_N]^T \quad (2-104)$$

Untuk prediksi dari persamaan *state* dan sinyal keluaran terlihat pada Persamaan (2-105) dan (2-106).

$$x(k_i + m | k) = A^m x(k_i) + \sum_{i=0}^{m-1} A^{m-i-1} BL(i)^T \eta \quad (2-105)$$

$$y(k_i + m | k) = CA^m x(k_i) + \sum_{i=0}^{m-1} CA^{m-i-1} BL(i)^T \eta \quad (2-106)$$

b. Cost Function

Pada perancangan kontroler prediktif dengan sistem yang memiliki lebih dari satu masukan dan keluaran. *Cost function* akan dirumuskan menggunakan fungsi Laguerre untuk memudahkan perhitungan. Pada Persamaan (2-107) terlihat rumus *cost function* dalam sistem SISO. Jika Y dan Δu merupakan vektor maka Persamaan (2-107) akan menjadi Persamaan (2-108).

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T R \Delta u \quad (2-107)$$

$$J = \sum_{m=1}^{N_p} (r(k_i) - y(k_i + m | k_i))^T (r(k_i) - y(k_i + m | k_i)) + \eta^T R_L \eta \quad (2-108)$$

Dimana R_L merupakan matrik diagonal ($N \times N$) dengan $r_w \geq 0$ pada diagonalnya. *Cost function* ini bertujuan untuk meminimalkan nilai kesalahan antara sinyal *set point* dan sinyal keluaran. Untuk mencari nilai η yang dapat meminimalkan *cost function* maka Persamaan (2-108) dapat ditulis menjadi Persamaan (2-109).

$$J = \sum_{m=1}^{N_p} (x(k_i + m | k_i))^T Q (x(k_i + m | k_i)) + \eta^T R_L \eta \quad (2-109)$$

Dimana matrik bobot $Q \geq 0$ dan $R_L > 0$. Q akan mempunyai dimensi yang sama dengan jumlah *state* dan R_L memiliki dimensi yang sesuai dengan η .

c. Minimalkan Fungsi Objektif

Persamaan (2-59) dapat disederhanakan seperti yang terlihat pada Persamaan (2-110).

$$\begin{aligned} x(k_i + m | k) &= A^m x(k_i) + \sum_{i=0}^{m-1} A^{m-i-1} B L(i)^T \eta \\ &= A^m x(k_i) + \phi(m)^T \eta \end{aligned} \quad (2-110)$$

Dengan memasukkan Persamaan (2-110) ke Persamaan (2-109) didapatkan Persamaan (2-111).

$$J = \eta^T \Omega \eta + 2\eta^T \Psi x(k_i) + \sum_{m=1}^{N_p} x(k_i)^T (A^T)^m Q A^m (x(k_i)) \quad (2-111)$$

Untuk mendapatkan nilai minimum dari Persamaan (2-111), maka dilakukan turunan parsial pada fungsi J dan didapatkan Persamaan (2-112).

$$\frac{\partial J}{\partial \eta} = 2\Omega\eta + 2\Psi x(k_i) \quad (2-112)$$

Asumsikan bahwa matrik Ω memiliki invers, untuk $\frac{\partial J}{\partial \eta} = 0$ maka didapatkan Persamaan (2-113).

$$\eta = -\Omega^{-1}\Psi x(k_i) \quad (2-113)$$

Dimana:

$$\Omega = \left(\sum_{m=1}^{N_p} \phi(m)Q\phi(m)^T + R_L \right)$$

$$\Psi = \left(\sum_{m=1}^{N_p} \phi(m)QA^m \right)$$

d. Minimum Cost Function

Untuk mendapatkan *cost function* minimum, maka Persamaan (2-101) dapat ditulis seperti Persamaan (2-114).

$$J = P - R + \sum_{m=1}^{N_p} x(k_i)^T (A^T)^m QA^m (x(k_i)) \quad (2-114)$$

Dimana:

$$P = (\eta + \Omega^{-1}\Psi x(k_i))^T \Omega (\eta + \Omega^{-1}\Psi x(k_i))$$

$$R = x(k_i)^T \Psi^T \Omega^{-1} \Psi x(k_i)$$

Dengan menggunakan solusi optimal dari Persamaan (2-113), maka didapatkan *cost function* minimum sesuai dengan Persamaan (2-115).

$$J_{\min} = x(k_i)^T \left(\sum_{m=1}^{N_p} x(k_i)^T (A^T)^m QA^m (x(k_i)) - \Psi^T \Omega^{-1} \Psi \right) x(k_i) \quad (2-115)$$

e. Receding Horizon Control

Sesuai dengan Persamaan (2-102) akan diambil elemen pertama saja dan didapatkan seperti Persamaan (2-116).

$$\Delta u(k_i) = L(0)^T \eta \quad (2-116)$$

Dimana fungsi Laguerre dengan memasukkan a dan N , $L(0)^T$ dapat dinyatakan dalam Persamaan (2-117).

$$L(0)^T = \sqrt{1-a^2} \begin{bmatrix} 1 & -a & a^2 & -a^3 & \dots & (-1)^{N-1} a^{N-1} \end{bmatrix} \quad (2-117)$$

Dengan η menyatakan *state* dari sistem yaitu $x(k_i)$, maka Persamaan (2-116) dapat dituliskan seperti Persamaan (2-118).

$$\Delta u(k) = -K_{mpc} x(k) \quad (2-118)$$

Dimana *gain state feedback* $K_{mpc} = L(0)^T \Omega^{-1} \Psi$.

2.9 Root Mean Square Error (RMSE)

RMSE merupakan suatu indikator kesalahan yang didasarkan pada kuadratis dari simpangan antara hasil model dengan hasil observasi yang dilakukan. Rumus dari RMSE terlihat pada Persamaan (2-119), dimana RMSE didapatkan dari jumlahan semua nilai kesalahan dibagi dengan banyaknya data yang diamati.

$$RMSE = \sqrt{\frac{\sum_{m=1}^N e_m^2}{N}} \quad (2-119)$$



3.1. Gambaran Umum Sistem

Secara umum, sistem yang dirancang pada penelitian ini adalah sistem *supervisory control and data acquisition* (SCADA) dengan arsitektur yang terlihat pada Gambar 3. 1. Sistem ini terdiri dari dua *plant* yaitu motor DC dan PCT-100. Perangkat *Remote Terminal Unit* (RTU) yang digunakan berupa satu unit komputer yang berisikan kontroler PC based. Terdapat satu komputer yang dijadikan sebagai *Master Terminal Unit* (MTU) untuk menampilkan *Human Machine Interface* (HMI) dari keseluruhan proses yang diamati. Pada arsitektur ini, terdapat dua buah komputer *client* yang digunakan untuk mengakses data dari MTU. Komunikasi yang digunakan bervariasi dengan menggunakan jaringan *ethernet* dan *wireless*.

The diagram illustrates the SCADA system architecture. At the top, a 'Client HMI (WinCC)' is connected to a 'Master Terminal Unit (MTU) HMI (WinCC)' via 'Wireless LAN'. The MTU is connected to two 'Client HMI (WinCC)' units via 'Ethernet'. The MTU is also connected to two 'PC Based Control (LabVIEW)' units via 'Ethernet'. These control units are connected to 'Advantech USB 4716' and 'ADAM 5000' modules, which are then connected to the 'Feedback Modular Servo System - Motor Speed Control System' and 'PCT 100' respectively. The PCT 100 is further connected to a 'Level Control System', 'Flow Control System', and 'Temperature Control System'.

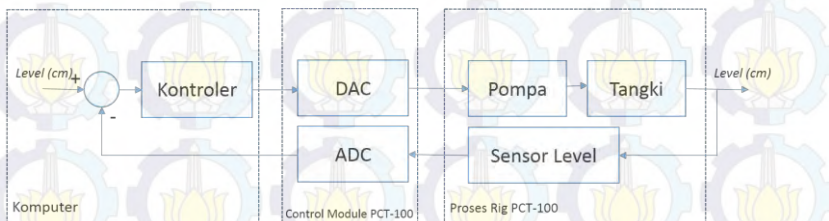
Gambar 3. 1. Arsitektur sistem SCADA

Pada *plant* motor DC dirancang sistem pengaturan kecepatan dengan menggunakan kontroler PI, dimana parameter kontroler didapat dengan menggunakan metode analitik. Perangkat akuisisi data yang digunakan adalah USB 4716 dengan *sampling rate* sebesar 200kS/s. Perangkat ini cocok digunakan pada motor DC yang memiliki dinamika perubahan parameter yang cepat. Terdapat dua mode operasi yang dirancang yaitu manual dan otomatis yang bertujuan untuk membedakan sistem tanpa kontroler dan menggunakan kontroler PI.

Penelitian ini memfokuskan pembahasan pada perancangan sistem pengaturan level dengan menggunakan kontroler PI dan prediktif. Perangkat akuisisi data yang digunakan adalah ADAM 5000L/TCP yang memiliki *sampling rate* sebesar 10S/s. Kontroler yang digunakan berupa PC based dengan menggunakan *software*.

3.2. Identifikasi Sistem

Proses identifikasi yang dilakukan adalah dengan memberikan sinyal uji pada masukan sistem dengan menggunakan *control module* PCT-100. Skema identifikasi terlihat pada Gambar 3. 2. Sinyal uji yang diberikan melalui program pada komputer yang dihubungkan menuju pompa melalui *control module*. Data sinyal keluaran dari sensor juga dimasukkan ke dalam komputer dalam bentuk data digital. Pengambilan data yang dilakukan adalah sebanyak lima kali dengan kondisi *valve* keluar ditutup.



Gambar 3. 2. Diagram blok identifikasi

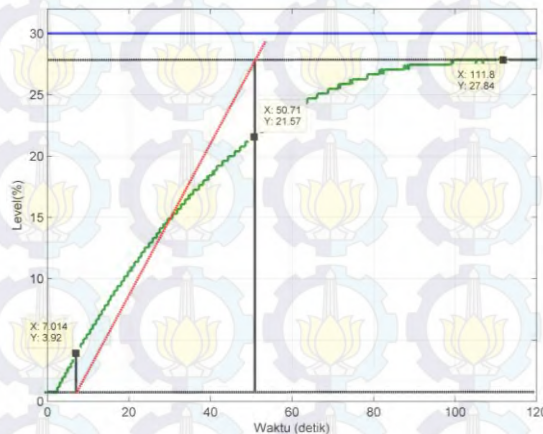
Dari hasil pengujian, dilakukan penarikan garis sesuai dengan metode Strejc untuk mendapatkan fungsi alih sistem. Pada Gambar 3. 3 terlihat respon sistem dengan penarikan garis Strejc. Sesuai dengan

prosedur yang telah dijelaskan pada Bab II, maka langkah selanjutnya adalah menghitung parameter sistem.

$$K = \frac{Y_{ss}}{X_{ss}} = \frac{30}{27.06} = 0.902$$

$$T_a = 50.71 - 7.014 = 43.696$$

$$T_u = 7.014$$



Gambar 3. 3. Penarikan garis Strejc pada respon

Maka didapatkan perbandingan nilai T_u dan T_a untuk mencari orde persamaan fungsi alih.

$$\frac{T_u}{T_a} = \frac{7.014}{43.696} = 0.16051$$

Dengan perbandingan nilai T_u dan T_a yang didapat maka mengacu pada Tabel 2. 1 didapatkan orde dari fungsi alih sebesar $n=2$. Selanjutnya dihitung nilai dari waktu konstan dan waktu tunda sistem.

$$T = \frac{T_a}{2.718} = 16.0765 \text{ detik}$$

$$T_u' = \left(\frac{T_u}{T_a} \right)_{tabel} \times T_a = 0.104 \times 43.696 = 4.5443 \text{ detik}$$

$$L = T_u - T_u' = 7.014 - 4.5443 = 2.4697 \text{ detik}$$

Didapatkan model matematika berupa fungsi alih umpan balik tertutup:

$$G(s) = \frac{0.902e^{-2,4697s}}{(1+16.0765s)^2}$$

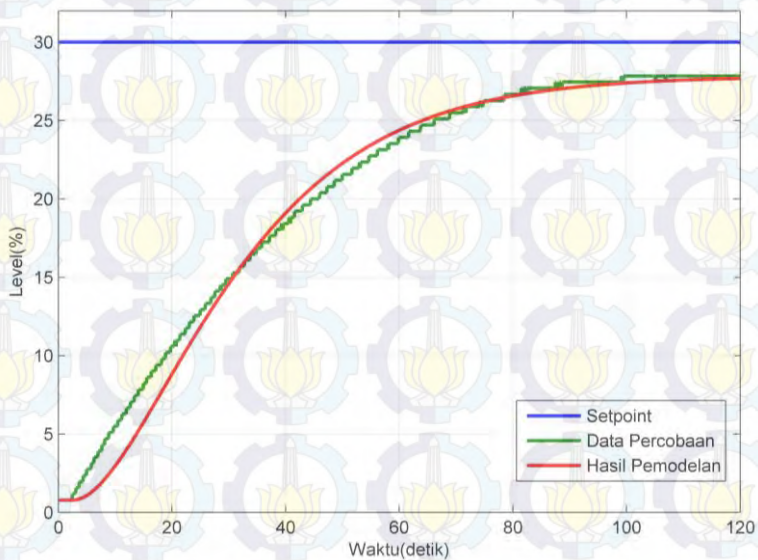
Pengambilan data dilakukan sebanyak lima kali, sehingga didapatkan fungsi alih sistem kontrol umpan balik tertutup seperti pada Tabel 3. 1.

Tabel 3. 1. Model matematis umpan balik tertutup

Percobaan	Model
1	$G(s) = \frac{0.902e^{-1.914s}}{(1+16.71s)^2}$
2	$G(s) = \frac{0.902e^{-1.4355s}}{(1+17.584s)^2}$
3	$G(s) = \frac{0.902e^{-3.6402s}}{(1+16.849s)^2}$
4	$G(s) = \frac{0.902e^{-3.9938s}}{(1+15.598s)^2}$
5	$G(s) = \frac{0.902e^{-2,4697s}}{(1+16.0765s)^2}$

3.2.1. Validasi Model

Untuk mengetahui persamaan yang telah diperoleh menggunakan metode Strejc dengan respon asli *plant*, maka dilakukan validasi untuk melihat kesesuaian model yang telah didapat. Parameter yang digunakan adalah RMSE untuk melihat respon yang paling dekat dengan respon asli. Semakin besar nilai RMSE yang didapat, maka semakin buruk model yang dibuat. Pada Gambar 3. 4 terlihat validasi dari model dari percobaan lima. Validasi yang dilakukan mengacu pada respon saat kondisi waktu tunak.



Gambar 3. 4. Validasi model

Pada Tabel 3. 2 terlihat perbandingan nilai RMSE dari masing-masing model. Dari hasil yang didapat model matematis yang diperoleh dari data pada percobaan lima mempunyai nilai RMSE yang paling kecil sebesar 4.78%. Hal tersebut menunjukkan bahwa fungsi alih tersebut memiliki respon yang paling mendekati hasil dari pengambilan data langsung.

Tabel 3. 2. Perbandingan nilai RMSE tiap model matematis

Percobaan	Model	RMSE
1	$G(s) = \frac{0.902e^{-1.914s}}{(1+16.71s)^2}$	6.64%
2	$G(s) = \frac{0.902e^{-1.4355s}}{(1+17.584s)^2}$	6.99%
3	$G(s) = \frac{0.902e^{-3.6402s}}{(1+16.849s)^2}$	9.001%
4	$G(s) = \frac{0.902e^{-3.9938s}}{(1+15.598s)^2}$	6.59%
5	$G(s) = \frac{0.902e^{-2.4697s}}{(1+16.0765s)^2}$	4.78%

3.2.2. Model Umpan Balik Terbuka

Setelah mendapatkan model matematis umpan balik tertutup, maka dicari model matematis untuk kondisi umpan balik terbuka. Sesuai dengan penurunan rumus pada Bab II, didapatkan parameter fungsi alih umpan balik terbuka. Model matematis umpan balik tertutup yang didapatkan terlihat pada Persamaan (3-1).

$$G(s) = \frac{0.902e^{-2.4697s}}{258.45s^2 + 32.15s + 1} \quad (3-1)$$

Dimana:

$$K' = 0.902$$

$$\alpha' = 258.45$$

$$\beta' = 32.15$$

Maka diperoleh parameter fungsi alih umpan balik terbuka sesuai dengan rumus yang telah diturunkan pada Bab II.

$$K = \frac{K'}{K_p(1-K')} = \frac{0.902}{1-0.902} = 9.2$$

$$\alpha = \alpha'(1 + K_p K) = 258.45(1 + 9.2) = 2636.19$$

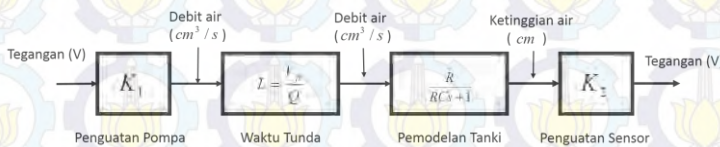
$$\begin{aligned}\beta &= \beta'(1 + K_p K) + K_p K \theta \\ &= 32.15(10.2) + 9.2(2.4697) \\ &= 350.65\end{aligned}$$

Dengan memasukkan parameter kedalam Persamaan (2-3), maka didapatkan fungsi alih seperti pada Persamaan (3-2).

$$G(s) = \frac{9.2e^{-2.4697s}}{2636.19s^2 + 350.65s + 1} \cong \frac{9.2(1 - 2.4697s)}{2636.19s^2 + 350.65s + 1} \quad (3-2)$$

3.3. Model Matematis Sistem

Dalam memodelkan sistem ini, terdapat beberapa bagian pemodelan seperti yang terlihat pada Gambar 3. 5. Sistem ini akan dimodelkan berdasarkan pengaruh dari komponen-komponen terhadap performa sistem. Komponen yang diamati adalah pompa, pipa yang menghubungkan pompa dengan tangki utama, sensor level, dan tangki utama.



Gambar 3. 5. Diagram pemodelan sistem

3.2.1 Mencari Nilai Parameter Sistem Saat Titik Kerja

Pada pengujian sistem dipilih titik kerja h_o sebesar 7.5 cm. Dilakukan pengujian dengan menghitung lama waktu pengosongan tangki dari titik kerja dengan membuka *valve* sebesar 100 % dan

didapatkan waktu sebesar 110 detik. Dari nilai yang didapat dilakukan perhitungan untuk mendapatkan debit air saat titik kerja.

$$Q_0 = \frac{\text{Volume tanki}}{\text{waktu}} = \frac{\frac{1}{4} \pi \times 16^2 \times 7.5}{110} = \frac{1507.96}{110} = 13.7 \text{ cm}^3 / \text{s}$$

a. Penguatan Pompa

Penguatan pompa ini disimbolkan sebagai K_1 yang merupakan rentang kerja tegangan masukan pada pompa. Diketahui rentang kerja pompa yang digunakan sebesar 0–10V dimana debit air yang keluar berbanding lurus dengan tegangan masukan sebesar 0–3.5lt / min, maka nilai K_1 dapat dirumuskan sesuai dengan Persamaan (3-3).

$$K_1 = \frac{3.5 \text{ lt} / \text{min} - 0}{10 \text{ V} - 0} = \frac{58.33 \text{ cm}^3 / \text{s} - 0}{10 \text{ V} - 0} = 5.83 \text{ cm}^3 / \text{sV} \quad (3-3)$$

b. Perhitungan Waktu tunda

Pada pemodelan ini dipilih Q_{in1} saat ujicoba sebesar 1 lt/min, maka didapatkan waktu tunda yang terjadi dan diperoleh fungsi alih waktu tunda seperti pada Persamaan (3-4).

$$L = \frac{V_p}{Q_{in1}} = \frac{\frac{1}{4} \pi \times 1^2 \times 90}{1 \text{ lt} / \text{min}} = \frac{70.68}{16.67} = 4.23 \text{ s}$$

$$\frac{Q_{in}(s)}{Q_{in1}(s)} = e^{-4.23s} \quad (3-4)$$

c. Pemodelan Tanki

Dilakukan perhitungan nilai R, sesuai dengan Persaman (2-7) dengan memasukkan parameter pada titik kerja. Dan diperoleh fungsi alih tanki seperti Persamaan (3-5) yang menyatakan hubungan antara ketinggian air dengan debit air pada tanki.

$$R = \frac{h_0}{Q_0} = \frac{7.5cm}{13.7cm^3/s} = 0.54s/cm^2$$

$$A = \frac{1}{4}\pi \times d^2 = \frac{1}{4}\pi \times 16^2 = 201.06cm^2$$

$$\frac{H(s)}{Q(s)} = \frac{0.54}{108.57s + 1} \quad (3-5)$$

d. Penguatan Sensor

Penguatan sensor ini disimbolkan sebagai K_2 yang merupakan rentang kerja tegangan masukan pada sensor. Diketahui rentang kerja pompa yang digunakan sebesar 0-10V dimana ketinggian air yang terbaca sebesar 0-18cm, maka nilai K_2 dapat dirumuskan sesuai dengan Persamaan (3-6).

$$K_1 = \frac{10V - 0}{18cm - 0} = 0.56V/cm \quad (3-6)$$

3.2.2 Menghitung Fungsi Alih Sistem

Dengan menggabungkan fungsi alih dari masing-masing kondisi yang telah didapat sebelumnya, maka didapatkan fungsi alih total seperti pada Persamaan (3-7).

$$\begin{aligned} G(s) &= K_1 \times \frac{Q_{in}(s)}{Q_{inl}(s)} \times \frac{H(s)}{Q_{in}(s)} \times K_2 \\ &= 5.83 \times e^{-4.23s} \times \frac{0.54}{108.57s + 1} \times 0.56 \\ &= \frac{1.76e^{-4.23s}}{108.57s + 1} \quad (3-7) \end{aligned}$$

3.2.3 Pendekatan Deret Taylor

Sistem memiliki waktu tunda yang kecil yaitu sebesar 4.23 detik dari pengujian 1000 detik, maka digunakan pendekatan deret Taylor seperti pada Persamaan (2-27) untuk mengubah waktu tunda dalam

bentuk fungsi alih sesuai dengan Persamaan (3-8). Maka didapatkan fungsi alih sistem yang baru seperti pada Persamaan (3-10).

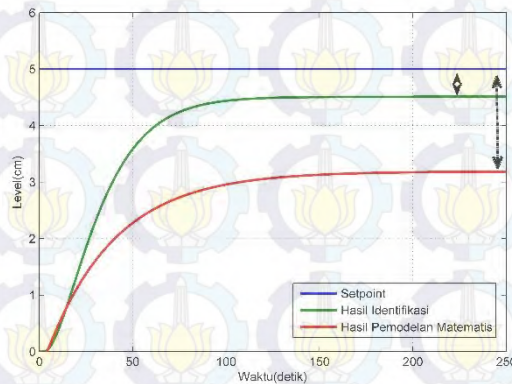
$$e^{-4.23s} \cong 1 - 4.23s \quad (3-8)$$

$$G(s) = \frac{1.76}{108.57s + 1} \times 1 - 4.23s \quad (3-9)$$

$$G(s) = \frac{-7.44s + 1.76}{108.57s + 1} \quad (3-10)$$

3.4. Pengujian Respon Umpan Balik Tertutup Tanpa Kontroler

Dari fungsi alih hasil identifikasi dan penurunan berdasarkan hukum fisis, maka dilakukan ujicoba umpan balik tertutup untuk melihat performansi sistem. Respon dari sistem terlihat pada Gambar 3. 6. Dari hasil pengujian sistem kontrol umpan balik tertutup didapatkan bahwa respon memiliki kesalahan sebesar 36.34% untuk pemodelan hasil identifikasi dan kesalahan sebesar 9.8% untuk pemodelan dari penurunan matematis. Sistem ini juga memiliki waktu tunda, maka dibutuhkan suatu kontroler untuk membuat respon sistem memiliki kesalahan nol serta dapat menghilangkan efek dari waktu tunda.



Gambar 3. 6. Respon umpan balik terbuka sistem

3.5. Perancangan Kontroler PI

Dari fungsi alih yang didapatkan pada berdasarkan hasil identifikasi dan pemodelan matematis, terlihat bahwa sistem hasil identifikasi menyerupai fungsi alih SPODT dan fungsi alih hasil penurunan model matematis seperti FOPDT. Perancangan kontroler PI yang akan dilakukan adalah berdasarkan metode *Internal Model Control* (IMC)[9]. Pertama akan dilakukan perancangan terhadap model hasil identifikasi berupa SOPDT. Dalam perancangan kontroler PI, dilakukan pendekatan SOPDT menjadi FOPDT sesuai dengan karakteristik respon berdasarkan besarnya koefisien redaman[9]. Pada Tabel 3. 3 terlihat parameter sistem dengan pendekatan FOPDT. Pendekatan FOPDT yang dilakukan mengacu pada Persamaan (3-11).

$$G(s) = \frac{K_m}{1 + \tau_m} e^{-D_m s} \quad (3- 11)$$

Tabel 3. 3. Parameter pendekatan SOPDT

Parameter	<i>Critically Damped</i>	<i>Overdamped</i>	<i>Underdamped</i>
Gain	$K_m = K_p$	$K_m = K_p$	$K_m = K_p$
Waktu Konstan	$\tau_m = 1.641\tau$	$\tau_m = [0.828 + 0.812 (\tau_{p1} / \tau_{p2}) + 0.172e^{-6.9\tau_{p2} / \tau_{p2}} \tau_{p1}]$	$\tau_m = 2\xi\tau$
Waktu Tunda	$D_m = 0.505\tau + D$	$D_m = \frac{1.116\tau_{p2}\tau_{p1}}{\tau_{p1} + 1.208\tau_{p1}} + D$	$D_m = \frac{\tau}{2\xi} + D$

Pendekatan yang dilakukan adalah pada fungsi alih umpan balik tertutup yang memiliki karakteristik *critically damped*. Maka didapatkan parameter FOPDT. Maka Persamaan (3-1) dapat dituliskan dalam bentuk FOPDT seperti pada Persamaan (3-12).

$$K_m = K_p = 0.92$$

$$\tau_m = 1.641\tau = 26.37$$

$$D_m = 0.505\tau + D = 10.58$$

$$G(s) = \frac{0.92}{1 + 26.37} e^{-1.058s} \quad (3-12)$$

Selanjutnya dilakukan perancangan kontroler PI terhadap dua fngsi alih yang diperoleh yaitu hasil identifikasi dan penurunan model matematis. Pada Tabel 3. 4 terlihat parameter kontroler PI dengan menggunakan metode *tuning* IMC. Spesifikasi respon dari perancangan kontroler ini adalah menghilangkan kesalahan waktu tunak dan *maksimum overshoot* sebesar 20 %.

Tabel 3. 4. Parameter kontroler PI

Kontroler	K_p	τ_i
PI	$\frac{2\tau_m + D_m}{2K_p\lambda}$	$\tau_m + 0.5D_m$

Dimana $\lambda = \max(0.25D_m, 0.2\tau_m) = 5.274$

Untuk model hasil identifikasi didapatkan nilai K_p dan τ_i seperti pada Persamaan (3-13) dan (3-14). Dan didapatkan fungsi alih dari kontroler PI seperti pada Persamaan (3-15).

$$K_p = \frac{2\tau_m + D_m}{2K_p\lambda} = \frac{2(0.92) + 10.58}{2(0.92)(5.274)} = 6.66 \quad (3-13)$$

$$\tau_i = \tau_m + 0.5D_m = 31.66 \quad (3-14)$$

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_i s} \right) = 6.66 \left(1 + \frac{1}{31.66s} \right) \quad (3-15)$$

Untuk model hasil penurunan model matematis didapatkan nilai K_p dan τ_i seperti pada Persamaan (3-16) dan (3-17). Dan didapatkan fungsi alih dari kontroler PI seperti pada Persamaan (3-18). Dimana $\lambda = \max(0.25D_m, 0.2\tau_m) = 21.714$

$$K_p = \frac{2\tau_m + D_m}{2K_p \lambda} = \frac{2(1.76) + 4.23}{2(1.76)(21.714)} = 2.89 \quad (3-16)$$

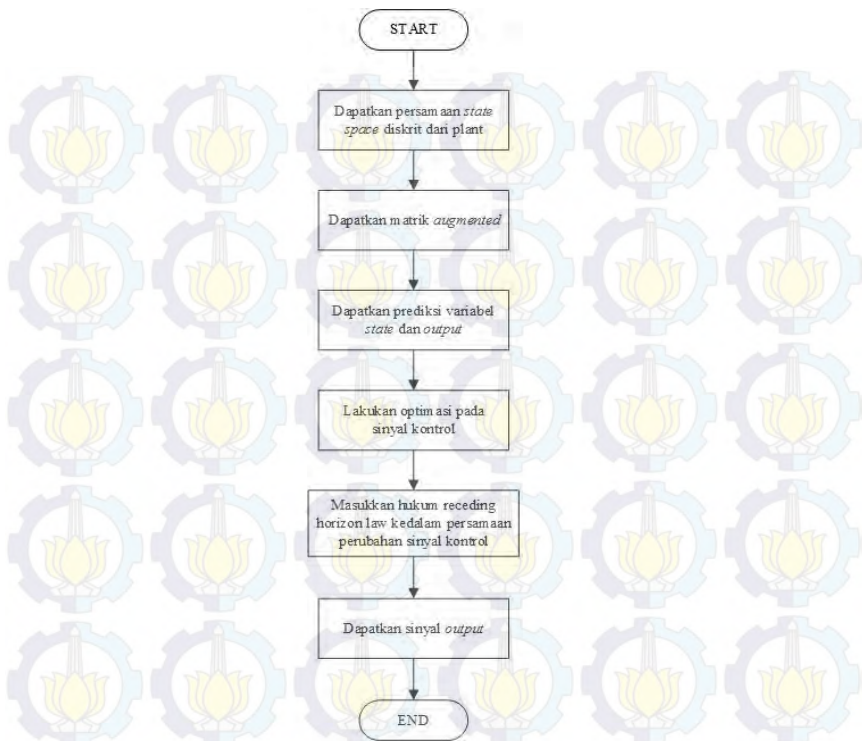
$$\tau_i = \tau_m + 0.5D_m = 110.685 \quad (3-17)$$

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_i s} \right) = 2.89 \left(1 + \frac{1}{110.685s} \right) \quad (3-18)$$

3.6. Perancangan Kontroler MPC

Dari penjabaran metode yang telah dijelaskan pada Bab II, dapat direpresentasikan dalam *flowchart* seperti yang terlihat pada Gambar 3. 7 dan didapatkan langkah-langkah pengerjaan sebagai berikut:

1. Dapatkan persamaan *state space* dari fungsi alih *plant*.
2. Dapatkan matrik augmented (A_e , B_e , C_e , dan D_e) yang akan dijadikan dasar dalam merumuskan prediksi *state* dan keluaran sistem.
3. Tentukan nilai dari N_p (*prediction horizon*) dan N_c (*control horizon*) untuk menentukan banyaknya *state* yang diprediksi dan sinyal kontrol.
4. Dapatkan matrik F dan Φ sebagai matrik yang merepresentasikan *state* terprediksi dan sinyal keluaran.
5. Lakukan optimasi dengan cara meminimumkan *cost function* dan dapatkan nilai sinyal keluaran.



Gambar 3. 7. Flowchart dari kontroler MPC

Perancangan kontroler MPC yang dilakukan berdasarkan fungsi alih yang didapat dari penurunan model matematis berdasarkan hukum fisis. Dari fungsi alih yang telah didapatkan pada Persamaan (3-7) akan diubah menjadi bentuk *state space* seperti pada Persamaan (3-19).

$$\begin{aligned}
 A_d &= [-0.0092] \\
 B_d &= [1] \\
 C_d &= [0.0168] \\
 D_d &= [-0.0685]
 \end{aligned}
 \tag{3-19}$$

State space pada Persamaan (3-19) masih dalam bentuk kontinyu, sementara dalam algoritma prediktif persamaan terlebih dahulu diubah ke dalam bentuk diskrit. Dengan memilih waktu sampling sebesar 0.1 detik, maka didapatkan *state space* diskrit sesuai dengan Persamaan (3-20).

$$\begin{aligned} A_p &= [0.9992] \\ B_d &= [0.1] \\ C_d &= [0.0168] \\ D_d &= [-0.0685] \end{aligned} \quad (3-20)$$

Untuk merancang kontroler prediktif maka akan dicari matrik *augmented* sebagai dasar dari perancangan. Pada Persamaan (3-21) didapatkan besar dari *matrik augmented*.

$$\begin{aligned} A_{-e} &= \begin{bmatrix} 0.9991 & 0 \\ 0.0542 & 1 \end{bmatrix} \\ B_{-e} &= \begin{bmatrix} 0.1 \\ 0.0054 \end{bmatrix} \\ C_{-e} &= [0 \quad 1] \end{aligned} \quad (3-21)$$

Dalam simulasi dan implementasi yang dilakukan dipilih nilai dari N_p sebesar 100 dan N_c sebesar 10. Dengan nilai tersebut maka didapatkan matrik F dengan dimensi 100x2 dengan 100 menyatakan besarnya N_p dan 2 menyatakan banyak *state*. Sementara pada matrik Phi akan berdimensi 100x10 dengan 10 menyatakan besarnya N_c . Dari matrik F dan Phi maka akan didapatkan nilai sinyal keluaran sesuai dengan Persamaan (2-58).

3.7. Perancangan SPC

Perancangan dari kontroler SPC dapat direpresentasikan pada *flowchart* sesuai dengan Gambar 3. 8. Langkah perancangan diawali dengan menurunkan persamaan matematis dari *plant* dan kontroler PI pada level *regulatory*. Diketahui dari persamaan *plant* yang didapat maka diubah ke dalam bentuk *state space* diskrit seperti pada Persamaan (3-22). Perancangan ini mengikuti SPC konfigurasi *cascade*.

$$\begin{aligned} A_d &= [0.9991] \\ B_d &= [0.1] \\ C_d &= [0.0542] \\ D_d &= [-0.2205] \end{aligned} \quad (3-22)$$

Persamaan kontroler yang didapat juga diubah ke dalam bentuk *state space* diskrit seperti pada Persamaan (3-23).

$$\begin{aligned} A_c &= [1] \\ B_c &= [0.1] \\ C_c &= [0.0154] \\ D_c &= [1.6680] \end{aligned} \quad (3-23)$$

Dari nilai yang telah didapat pada Persamaan (3-22) dan (3-23), maka dihitung persamaan matrik sesuai pada Persamaan (2-79) dan (2-80) dan didapatkan Persamaan (3-24) dan (3-25).

$$A_p = \begin{bmatrix} 0.9901 & 0.0015 \\ -0.0054 & 1 \end{bmatrix} \quad (3-24)$$

$$B_p = \begin{bmatrix} 1.6680 & 0.1 \\ 0.1 & 0 \end{bmatrix}$$

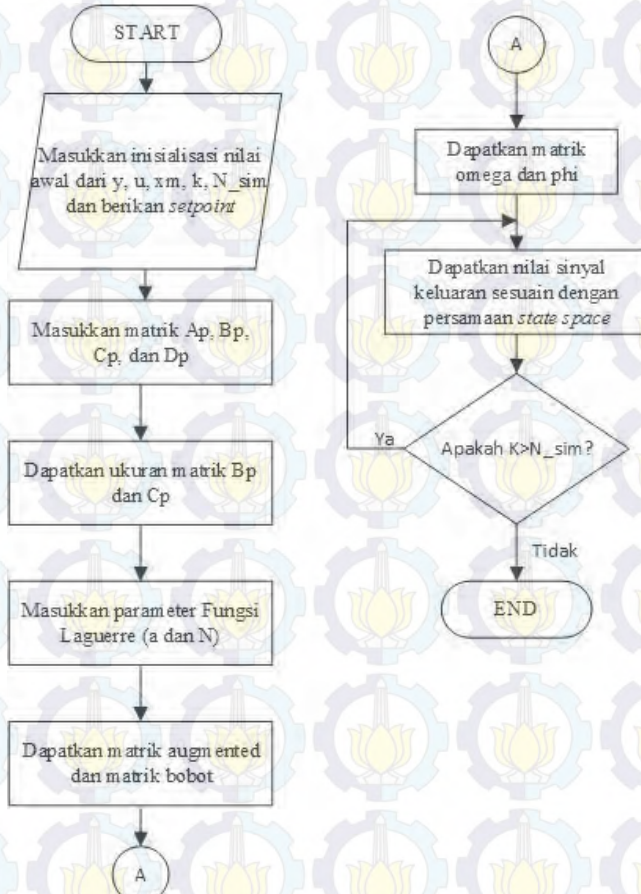
$$C_p = \begin{bmatrix} 0.0542 & 0 \\ -0.0904 & 0.0154 \end{bmatrix} \quad (3-25)$$

$$D_p = \begin{bmatrix} 0 & 0 \\ 1.6680 & 1 \end{bmatrix}$$

Dengan memilih banyaknya simulasi $N_{sim}=1000$. Dipilih parameter fungsi Laguerre yaitu $a=0.5$ dan $N=5$ serta besarnya *prediction horizon* sebesar 5. Dan didapatkan nilai dari sinyal keluaran seperti Pada Persamaan (3-26). Perhitungan matematis secara rinci dijabarkan pada bagian Lampiran.

$$y = \begin{bmatrix} y_k \\ u_k \end{bmatrix} \quad (3-26)$$

$$= \begin{bmatrix} 1.97 \times 10^{-4} & 5.89 \times 10^{-4} & 0.0012 & 0.002 & 0.0029 \\ -2.95 \times 10^{-4} & -8.82 \times 10^{-4} & -0.0018 & -0.0029 & -0.0044 \end{bmatrix}$$



Gambar 3. 8. Flowchart algoritma SPC

3.8. Kestabilan Sistem dengan Waktu Tunda

Suatu sistem dikatakan stabil jika memiliki nilai masukan terbatas dan nilai keluaran terbatas [17]. Pada sistem LTI (*linear time invariant*), kestabilan dari sistem dapat dilihat dari akar-akar persamaan fungsi alih. Jika semua akar-akar berada di sebelah kiri sumbu imajiner, maka sistem dikatakan stabil. Sistem tidak stabil jika terdapat minimal satu akar yang berada di sebelah kanan sumbu imajiner. Sistem dikatakan stabil marginal jika semua akar berada pada sumbu imajiner [10].

Pada bagian ini, akan dibahas mengenai efek dari waktu tunda terhadap kestabilan dari sistem. Analisa dilakukan dalam dua cara yaitu menggunakan diagram Bode dan Nyquist. Waktu tunda akan dimodelkan dalam domain frekuensi seperti terlihat pada Persamaan (3-27).

$$G(j\omega) = e^{-j\omega T} \quad (3-27)$$

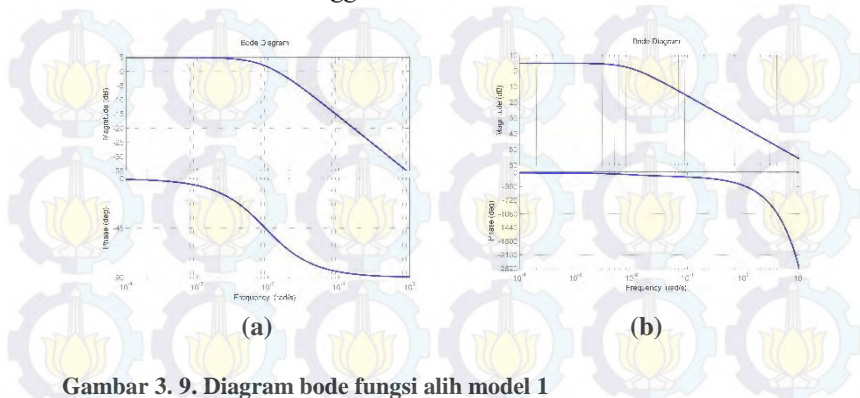
Dimana T merupakan besar waktu tunda yang terjadi. Magnitudo dan fasa dari persamaan tersebut dapat dirumuskan seperti pada Persamaan (3-28) dan (3-29). Dalam fungsi alih waktu tunda ini memiliki magnitude yang tetap yaitu sebesar 1. Jika direpresentasikan dalam diagram Bode maka akan memiliki magnitudo sebesar 0 dB.

$$|G(j\omega)| = |\cos \omega T - j \sin \omega T| = 1 \quad (3-28)$$

$$\angle G(j\omega) = -\omega T \quad (\text{radian}) \quad (3-29)$$

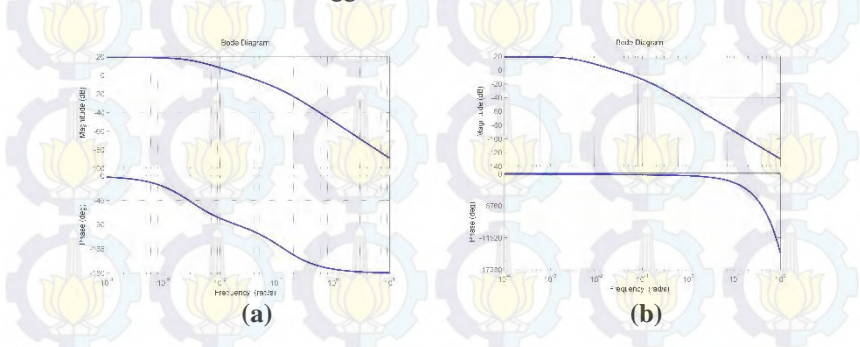
Dalam analisa terhadap domain frekuensi, maka akan digunakan diagram Bode untuk melihat respon magnitudo dan fasa dari sistem. Pada Gambar 3. 9 terlihat diagram bode dari fungsi alih hasil penurunan berdasarkan hukum fisika (Model 1). Perbandingan antara fungsi alih tanpa waktu tunda dan dengan waktu tunda terlihat pada Gambar 3. 9 (a) dan (b). Sebelum ditambahkan waktu tunda respon memiliki *phase margin* sebesar 2.18 radian. Dimana *phase margin* merupakan fasa pada saat respon magnitudo mencapai nilai 0 dB. Setelah ditambahkan waktu tunda sistem memiliki *phase margin* sebesar 2.11 radian dan gain margin sebesar 27.3 dB. Dimana *gain margin* merupakan magnitudo pada saat respon fasa mencapai nilai -180 deg. Dari perbandingan diagram Bode terlihat bahwa penambahan waktu tunda tidak berdampak terhadap

magnitudo dari sistem, tetapi menimbulkan keterlambatan respon yang besar dalam frekuensi tinggi.



Gambar 3. 9. Diagram bode fungsi alih model 1

Pada Gambar 3. 10 terlihat diagram Bode dari fungsi alih hasil identifikasi sistem dengan metode Strejc. Gambar (a) menunjukkan diagram Bode dari fungsi alih sebelum ditambahkan waktu tunda dan memiliki *phase margin* sebesar 1.48 radian. Sementara setelah ditambahkan waktu tunda dalam fungsi alih sistem terlihat pada Gambar (b), sistem memiliki *gain margin* sebesar 24.2 dB dan *phase margin* sebesar 1.41 radian. Sama seperti model sebelumnya, penambahan waktu tunda tidak berpengaruh terhadap respon magnitudo sistem. Pada respon fasa, penambahan waktu tunda akan menyebabkan keterlambatan yang besar dalam frekuensi tinggi.

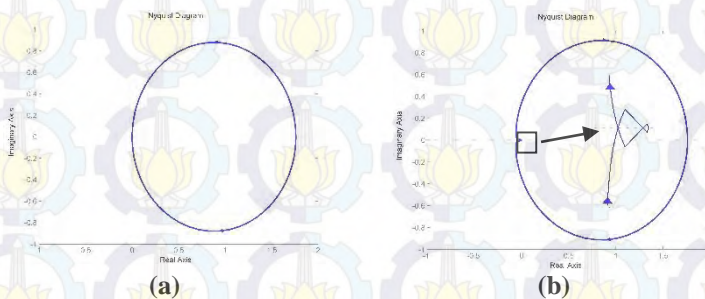


Gambar 3. 10. Diagram bode fungsi alih model 2

Selanjutnya dilakukan analisa kestabilan dengan menggunakan kriteria Nyquist. Terdapat tiga kriteria kestabilan nyquist yaitu [10]:

1. Tidak ada *locus* yang memutar pada titik $-1+j0$. Hal ini menandakan bahwa tidak ada pole dari $G(s)H(s)$ yang berada di sebelah kanan sumbu *imaginer*.
2. Terdapat satu atau lebih putaran *locus* berlawanan arah jarum jam pada titik $-1+j0$. Pada kasus ini, jika banyaknya putaran pada titik tersebut sama dengan jumlah pole yang berada di sebelah kanan sumbu imajiner maka sistem dikatakan stabil.
3. Terdapat satu atau lebih putaran *locus* searah jarum jam pada titik $-1+j0$. Pada kasus ini, sistem dikatakan tidak stabil.

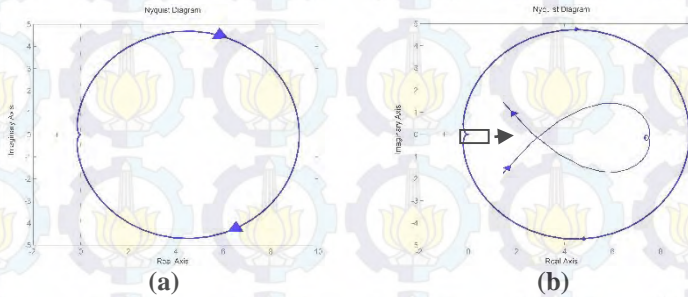
Pada Gambar 3. 11 terlihat perbandingan plot Nyquist antara model fungsi alih hasil penurunan berdasarkan hukum fisika tanpa waktu tunda dan dengan waktu tunda. Terlihat pada gambar (a) sebelum diberi penambahan persamaan waktu tunda, sistem dikatakan stabil absolut karena memiliki pole di sebelah kiri sumbu imajiner dan tidak ada *locus* yang memutar pada titik $-1+j0$. Setelah adanya waktu tunda terlihat pada Gambar (b), *locus* akan menjadi lebih dekat dengan titik $-1+j0$ yang membuat tingkat kestabilan sistem akan berkurang.



Gambar 3. 11. Nyquist fungsi alih model 1

Pada Gambar 3. 12 menunjukkan perbandingan plot Nyquist antara model fungsi alih hasil identifikasi sistem menggunakan metode Strejc tanpa waktu tunda dan dengan penambahan waktu tunda. Dari hasil yang didapat pada gambar (a) terlihat bahwa tidak ada *locus* yang memutar pada titik $-1+j0$. Hal ini menandakan bahwa tidak ada pole dari sistem

yang berada di sebelah kanan sumbu *imaginer*, sehingga dapat disimpulkan bahwa sistem stabil absolut. Setelah terjadi penambahan fungsi waktu tunda seperti terlihat pada gambar (b), locus akan menjadi lebih dekat dengan titik $-1+j0$ yang membuat tingkat kestabilan sistem berkurang.



Gambar 3. 12. Nyquist fungsi alih model 2

Tingkat kestabilan sistem dengan adanya waktu tunda dapat dikatakan stabil relatif yang bergantung dari jarak locus terhadap titik $-1+j0$ dan dibuktikan dalam *phase margin* yang secara matematis dapat ditulis seperti pada Persamaan (3-50).

$$\gamma = 180^\circ + \phi \quad (3-30)$$

Dimana γ merupakan fasa margin dan ϕ adalah fasa saat *frekuensi gain crossover*. Setelah penambahan fasa dari fungsi waktu tunda seperti pada Persamaan (3-49), maka *phase margin* sistem akan menjadi Persamaan (3-51).

$$\gamma = 180^\circ + \phi - \omega T \quad (3-31)$$

Dengan adanya waktu tunda, maka akan menyebabkan *phase margin* akan tertinggal sebesar ωT sehingga dapat mengurangi tingkat kestabilan sistem.



Halaman ini sengaja dikosongkan

BAB 4

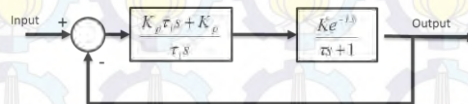
PENGUJIAN DAN ANALISIS

Bab ini berisi tentang hasil simulasi dan implementasi serta *analisis* dari perancangan yang sudah dilakukan pada Bab III. Pengujian akan dilakukan terhadap empat jenis kontroler yang telah dirancang yaitu kontroler PI, kontroler MPC level *regulatory*, SPC konfigurasi *cascade*, dan *parallel*. Pengujian kontroler terhadap dua model hasil identifikasi dan penurunan rumus matematis.

4.1. Simulasi Kontroler PI dan Analisis

4.1.1. Simulasi dengan Masukan Sinyal *Step*

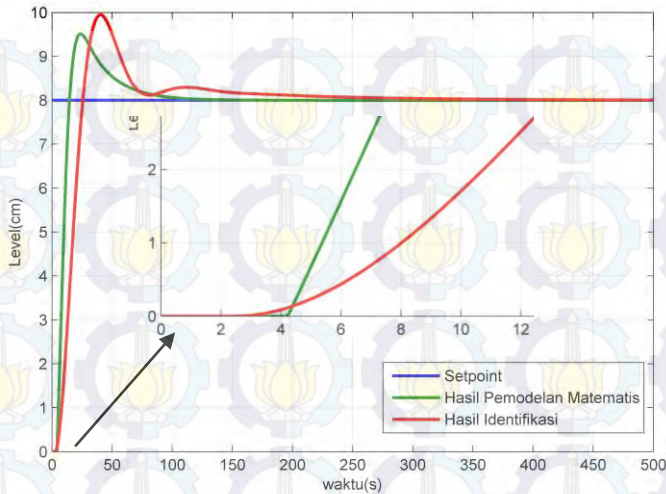
Pada simulasi ini, sistem dirancang dengan kontroler PI seperti terlihat pada Gambar 4. 1. Perbandingan parameter dari respon terlihat pada Tabel 4. 1. Simulasi dilakukan dengan memberikan masukan sinyal *step* terhadap masukan. Pada proses simulasi ini, dilakukan pengambilan data sebanyak 100 kali dari masing-masing percobaan dengan menggunakan *for loop*. Pada pengujian ini akan dibandingkan respon sistem terhadap masukan dengan menggunakan dua model yang berbeda yaitu hasil identifikasi dan penurunan model matematis. Pada Gambar 4. 2 terlihat perbandingan respon dari kedua fungsi alih *plant*.



Gambar 4. 1. Diagram blok simulasi PI

Tabel 4. 1. Perbandingan respon kontroler PI

Model	Overshoot	Waktu Konstan	Ts ($\pm 2\%$)
Hasil Identifikasi	24.23%	17.8 detik	157.4 detik
Hasil Pemodelan Matematis	18.87%	10 detik	78.4 detik



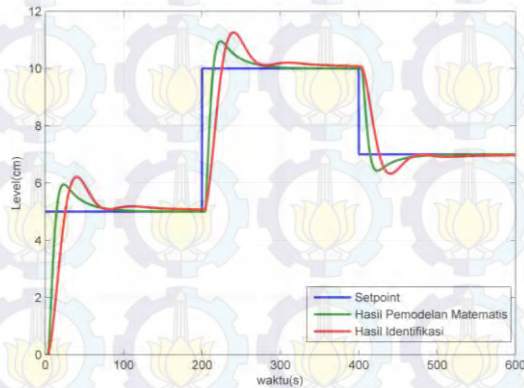
Gambar 4. 2. Respon sistem dengan kontroler PI

Dari hasil yang didapatkan terlihat bahwa respon dengan waktu konstan yang lebih cepat terjadi jika menggunakan model hasil penurunan matematis dengan memiliki τ sebesar 10 detik. Terjadi perbedaan dari respon yang dihasilkan dikarenakan, saat penurunan model matematis terdapat beberapa kondisi yang diabaikan salah satunya yaitu tidak diperhitungkan pengkondisian sinyal yang terdapat pada aktuatur dan sensor. Overshoot yang didapat dengan menggunakan model hasil identifikasi sebesar 24.23% dan nilai tersebut melebihi spesifikasi design yang ditentukan. Hal ini disebabkan oleh metode tuning PI yang digunakan tidak sesuai dengan model yang digunakan. Dalam tuning PI IMC menggunakan acuan dari model umpan balik tertutup.

4.1.2. Simulasi dengan Variasi Set Point

Pada pengujian ini, kontroler PI diberi masukan yang bervariasi dalam waktu yang berkelanjutan. Pengujian ini dilakukan untuk melihat efek dari waktu tunda pada sistem dengan menggunakan kontroler PI. Pada Gambar 4. 3 terlihat respon sistem dengan menggunakan kontroler PI saat diberi masukan berupa variasi *set point*. Hasil yang didapatkan

adalah kontroler PI tidak mampu menghilangkan efek dari waktu tunda yang terjadi akibat bentuk fisik *plant*. Namun kontroler PI dengan menggunakan model hasil identifikasi maupun berdasarkan penurunan rumus matematis mampu menghilangkan e_{ss} dengan waktu sebesar 157.4 detik dan 78.4 detik.



Gambar 4. 3. Respon kontroler PI terhadap masukan variasi *set point*

4.2. Simulasi Kontroler MPC Level *Regulatory* dan Analisis

4.2.1. Simulasi dengan Masukan Sinyal *Step*

Pada simulasi ini, sistem dirancang dengan kontroler MPC level *regulatory* dengan masukan sinyal *step*. Gambar 4. 4 menunjukkan diagram blok kontroler prediktif. Pada konfigurasi ini, MPC digunakan pada level *regulatory* untuk menggantikan kontroler PI. Pada proses simulasi ini, dilakukan pengambilan data sebanyak 100 kali dari masing-masing percobaan dengan menggunakan *for loop*. Waktu sampling yang dipakai dalam simulasi adalah sebesar 0. 1s dengan total waktu simulasi sebesar 100s.

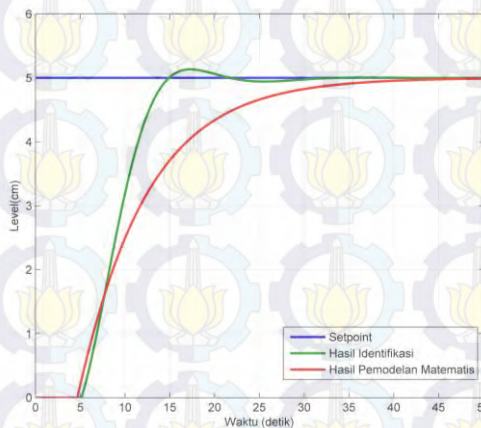


Gambar 4. 4.Blok diagram kontroler prediktif

Pada Gambar 4. 5 terlihat respon hasil simulasi kontroler MPC dengan masukan *unit step* pada dua model yang berbeda. Respon dengan model hasil identifikasi memiliki *overshoot* sebesar 0.64%. Pada Tabel 4. 2 terlihat perbandingan respon hasil identifikasi dengan penurunan model secara matematis. Pada pengujian ini kontroler MPC tidak mampu menghilangkan waktu tunda diakibatkan tidak memiliki informasi mengenai sinyal referensi yang akan datang. Algoritma kontroler prediktif mampu bekerja dengan baik jika sinyal referensi yang akan datang diketahui. Sinyal kontrol yang diprediksi akan membuat sinyal keluaran mengikuti sinyal referensi.

Tabel 4. 2. Perbandingan respon kontroler MPC

Model	τ	Overshoot
Identifikasi	4.85 detik	0.64%
Pemodelan Matematis	7.74 detik	-

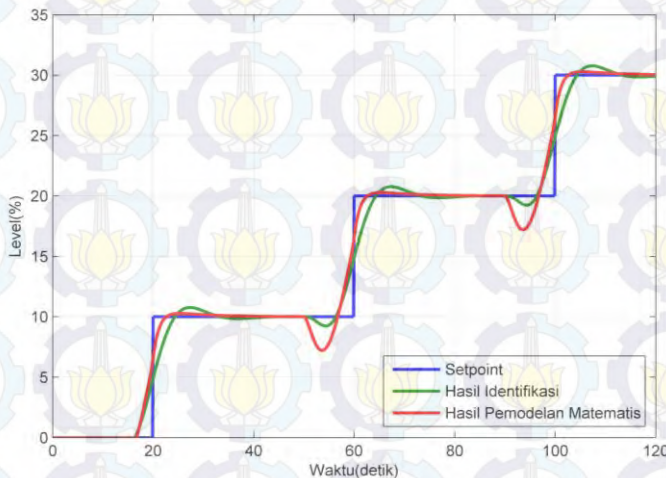


Gambar 4. 5. Respon sistem dengan kontroler MPC diberi masukan sinyal *step*

4.2.2. Simulasi MPC dengan Variasi *Set Point*

Pada pengujian ini, kontroler MPC diuji dengan diberi masukan berupa *set point* yang bervariasi dalam rentang waktu tertentu. Pengujian ini dilakukan untuk melihat efek waktu tunda yang dapat dihilangkan.

Informasi variasi *set point* yang dilakukan sudah diketahui pada kondisi awal. Pada Gambar 4. 6 terlihat hasil simulasi dan menunjukkan bahwa kontroler MPC mampu menghilangkan waktu tunda dengan memprediksi sinyal keluaran. Kontroler MPC sudah memprediksi sinyal keluaran 10 detik sebelum perubahan *set point* dilakukan. Model hasil penurunan rumus matematis, respon yang dihasilkan memiliki penurunan nilai sebesar 27.77% saat memprediksi sinyal keluaran, sementara model hasil identifikasi memiliki penurunan nilai sebesar 7.45%. Hal ini diakibatkan pemodelan dengan menggunakan penurunan hukum dasar fisika banyak mengabaikan faktor – faktor yang mempengaruhi sinyal keluaran sistem, salah satunya adalah pengkondisian sinyal pada pemodelan sensor dan aktuatur yang digunakan. Pendekatan waktu tunda yang dilakukan menggunakan pendekatan Taylor orde pertama.

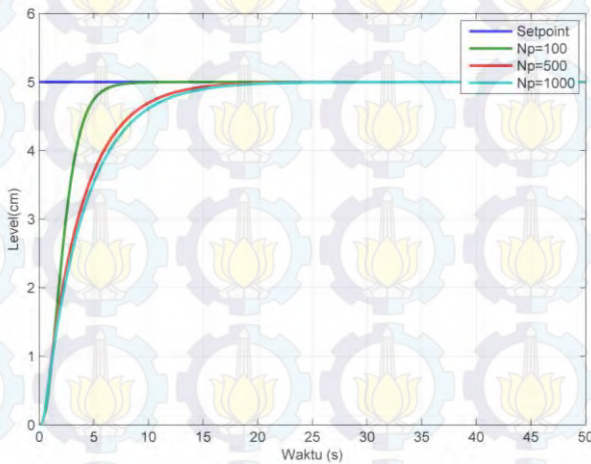


Gambar 4. 6. Pengujian kontroler MPC dengan variasi set point

4.2.3. Simulasi dengan Variasi N_p

Variasi pada N_p akan menghasilkan sinyal keluaran yang diprediksi akan berbeda karena banyaknya kondisi *state*. Semakin besar N_p semakin banyak sinyal kontrol yang diprediksi. Pada Gambar 4. 7 menunjukkan respon hasil dari simulasi dengan variasi N_p . Dari hasil didapatkan bahwa semakin besar nilai dari N_p yang digunakan maka

respon dari sistem akan mengalami peningkatan waktu konstan. Dari data yang didapat dengan N_p sebesar 1000 maka akan didapatkan waktu konstan sebesar 4.156 detik. Sementara, N_p sebesar 100 dan 500 mempunyai waktu konstan sebesar 2.572 detik dan 3.847 detik. Dari hasil ini, maka dapat disimpulkan bahwa semakin banyak *state* yang diprediksi maka nilai sinyal kontrol akan lebih lama mencapai nilai referensi yang diberikan. Nilai kesalahan yang didapat dihilangkan dengan waktu yang lebih lama.



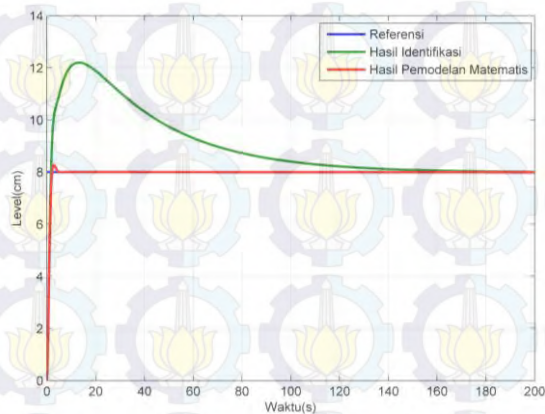
Gambar 4. 7. Respon sistem dengan kontroler MPC variasi N_p

4.3. Simulasi Kontroler SPC Konfigurasi *Cascade* dan Analisis

4.3.1. Simulasi dengan Masukan Sinyal *Step*

Pada simulasi ini, sistem dirancang dengan kontroler SPC konfigurasi *cascade* dan diberi masukan berupa sinyal *step*. Simulasi dilakukan dengan melakukan variasi pada nilai *set point*. Pada proses simulasi ini, dilakukan pengambilan data sebanyak 100 kali dari masing-masing percobaan dengan menggunakan *for loop*. Waktu sampling yang dipakai dalam simulasi adalah sebesar 0. 1s dengan total waktu simulasi sebesar 100s. Pada Gambar 4. 8 terlihat respon hasil simulasi kontroler SPC konfigurasi *cascade* dengan masukan sinyal *step*. Dari respon yang

didapat terlihat bahwa efek dari waktu tunda yang terjadi dapat dihilangkan akibat pendekatan dengan Pendekatan Taylor orde pertama. Pada Tabel 4. 3 terlihat perbandingan parameter respon dari kontroler SPC *cascade*. Respon dari sistem dengan model hasil identifikasi maupun penurunan rumus matematis memiliki waktu konstan yang cepat yaitu 1.026 detik dan 1.055 detik. Namun kontroler ini memiliki overshoot yang sangat besar saat digunakan model dari hasil identifikasi yaitu sebesar 52.5%. *Overshoot* yang didapat melebihi spesifikasi perancangan yang ditentukan. Hal ini disebabkan metode *tuning* parameter PI yang didapat menggunakan model sistem secara umpan balik tertutup.



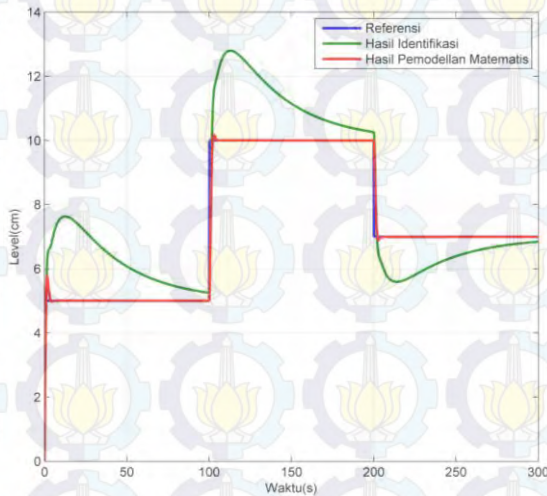
Gambar 4. 8. Respon sistem dengan kontroler SPC konfigurasi *cascade* diberi masukan sinyal *step*

Tabel 4. 3. Perbandingan respon kontroler SPC *cascade*

Model	<i>Overshoot</i>	Waktu Konstan	Ts ($\pm 2\%$)
Hasil Identifikasi	52.5%	1.026 detik	130.7 detik
Hasil Pemodelan Matematis	3.475%	1.055 detik	3.627 detik

4.3.2. Simulasi dengan Variasi Set Point

Pada pengujian ini, dilakukan variasi set point terhadap masukan sistem. Hal ini dilakukan untuk menguji sistem dalam menghilangkan efek dari waktu tunda yang terjadi akibat dari bentuk fisik *plant*. Pada Gambar 4. 9 terlihat respon sinyal keluaran sistem dengan masukan bervariasi. Terlihat bahwa respon yang dihasilkan mampu menghilangkan waktu tunda yang terjadi dengan pemodelan waktu tunda menggunakan pendekatan Taylor orde pertama.

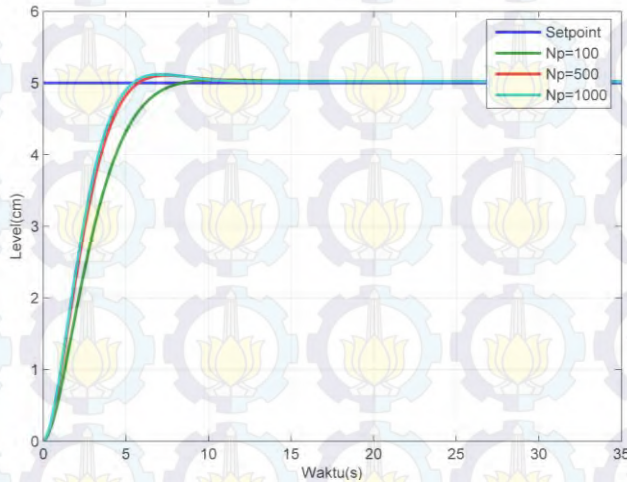


Gambar 4. 9. Respon sistem dengan kontroler SPC konfigurasi *cascade* dengan variasi *set point*

4.3.3. Simulasi dengan Variasi N_p

Variasi pada N_p akan menghasilkan sinyal keluaran yang diprediksi akan berbeda karena banyaknya kondisi *state*. Pada Gambar 4. 10 menunjukkan respon hasil dari simulasi dengan variasi N_p . Dari hasil didapatkan bahwa semakin besar nilai dari N_p yang digunakan maka respon dari sistem akan mengalami penurunan waktu konstan. Dari data yang didapat dengan N_p sebesar 1000 maka akan didapatkan waktu konstan sebesar 2.65 detik. Sementara, N_p sebesar 100 dan 500 mempunyai waktu konstan sebesar 3.33 detik dan 2.72 detik. Respon ini juga memiliki nilai overshoot yang berbeda yaitu untuk N_p sebesar 1000,

500, dan 100 berturut memiliki nilai yaitu 1.39 %, 1.27 %, dan 0.64 %. Dengan menggunakan kontroler SPC konfigurasi *cascade*, respon sistem yang dihasilkan mempunyai karakteristik respon yang lebih baik jika hanya menggunakan kontroler PI, hal ini terlihat pada hasil *time constant* dan *overshoot* yang didapat.



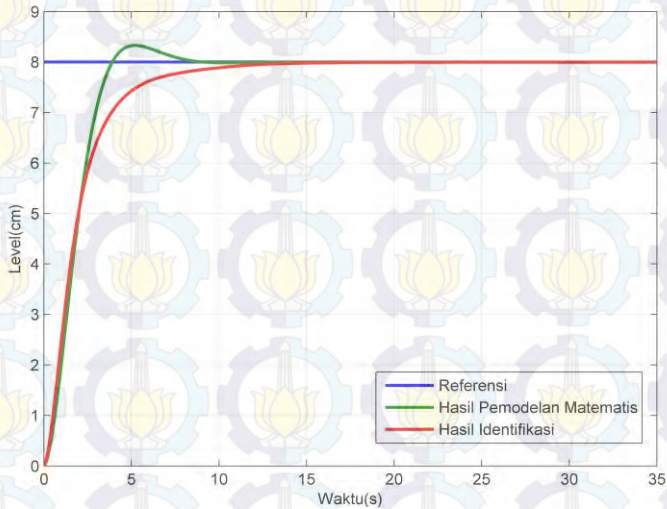
Gambar 4. 10. Respon sistem dengan kontroler SPC konfigurasi *cascade* dengan variasi N_p

4.4. Simulasi Kontroler SPC Konfigurasi *Parallel* dan Analisis

4.4.1. Simulasi dengan Masukan Sinyal *Step*

Pada simulasi ini, sistem dirancang dengan kontroler SPC konfigurasi *parallel* dengan masukan sinyal *step*. Pada proses simulasi ini, dilakukan pengambilan data sebanyak 100 kali dari masing-masing percobaan dengan menggunakan *for loop*. Waktu sampling yang dipakai dalam simulasi adalah sebesar 0.1s dengan total waktu simulasi sebesar 100s. Pada Gambar 4. 11 terlihat respon hasil simulasi kontroler SPC konfigurasi *parallel* dengan masukan *unit step*. Dari respon yang didapat terlihat bahwa efek dari waktu tunda yang terjadi dapat dihilangkan. Pada Tabel 4. 4 terlihat perbandingan parameter kontrol pada SPC *parallel*. Respon dengan model hasil pemodelan matematis memiliki *overshoot*

sebesar 3.31%, sementara respon dengan model hasil penurunan rumus matematis tidak memiliki *overshoot* dan memiliki respon orde pertama dengan waktu konstan sebesar 2.014 detik.



Gambar 4. 11. Respon sistem dengan kontroler SPC konfigurasi *parallel* diberi masukan sinyal *step*

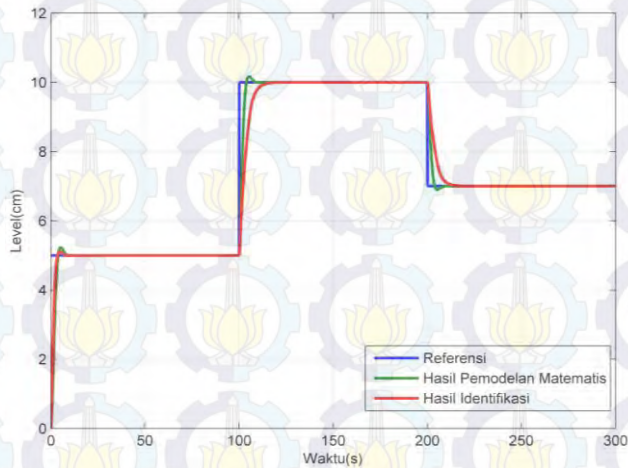
Tabel 4. 4. Perbandingan parameter respon pada kontroler SPC *parallel*

Model	Overshoot	Waktu Konstan	Tr
Hasil Identifikasi	-	2.014 detik	6.042 detik
Hasil Pemodelan Matematis	3.31%	2.026 detik	3.9 detik

4.4.2. Simulasi dengan Variasi Set Point

Pada pengujian ini, sistem dengan kontroler SPC konfigurasi *parallel* diberi masukan yang variasi. Hal ini dilakukan untuk menguji sistem dalam menghilangkan efek dari waktu tunda yang terjadi akibat

dari bentuk fisik *plant*. Pada Gambar 4. 12 terlihat respon sinyal keluaran sistem dengan masukan bervariasi. Terlihat bahwa respon yang dihasilkan mampu menghilangkan waktu tunda yang terjadi dengan pemodelan waktu tunda menggunakan pendekatan Taylor orde pertama. Perbandingan dilakukan dengan menggunakan dua model yang telah didapat pada Bab III.

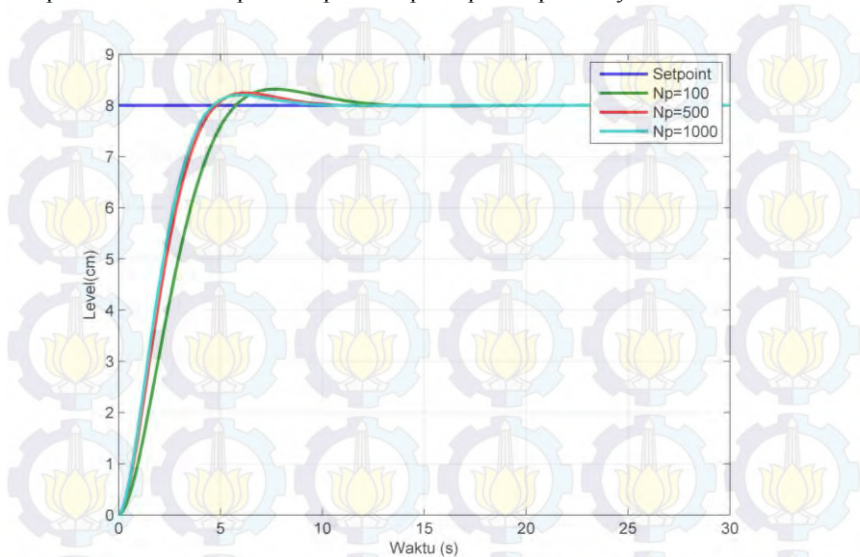


Gambar 4. 12. Respon sistem dengan kontroler SPC konfigurasi *parallel* dengan variasi *set point*

4.4.3. Simulasi dengan Variasi N_p

Variasi pada N_p akan menghasilkan sinyal keluaran yang diprediksi akan berbeda karena banyaknya kondisi *state*. Pada Gambar 4. 13 menunjukkan respon hasil dari simulasi dengan variasi N_p . Dari hasil didapatkan bahwa semakin besar nilai dari N_p yang digunakan maka respon dari sistem akan mengalami penurunan waktu konstan. Dari data yang didapat dengan N_p sebesar 1000 maka akan didapatkan waktu konstan sebesar 3.3 detik. Sementara, N_p sebesar 100 dan 500 mempunyai waktu konstan sebesar 5.5 detik dan 4.13 detik. Respon ini juga memiliki nilai *overshoot* yang berbeda yaitu untuk N_p sebesar 1000, 500, dan 100 berturut memiliki nilai yaitu 4.29 %, 3.73 %, dan 1.69 %. Nilai N_p akan mempengaruhi banyak sinyal kontrol yang diprediksi,

semakin besar nilai N_p maka semakin banyak pula sinyal kontrol yang diprediksi dan mampu mempu mempercepat respon sinyal keluaran.

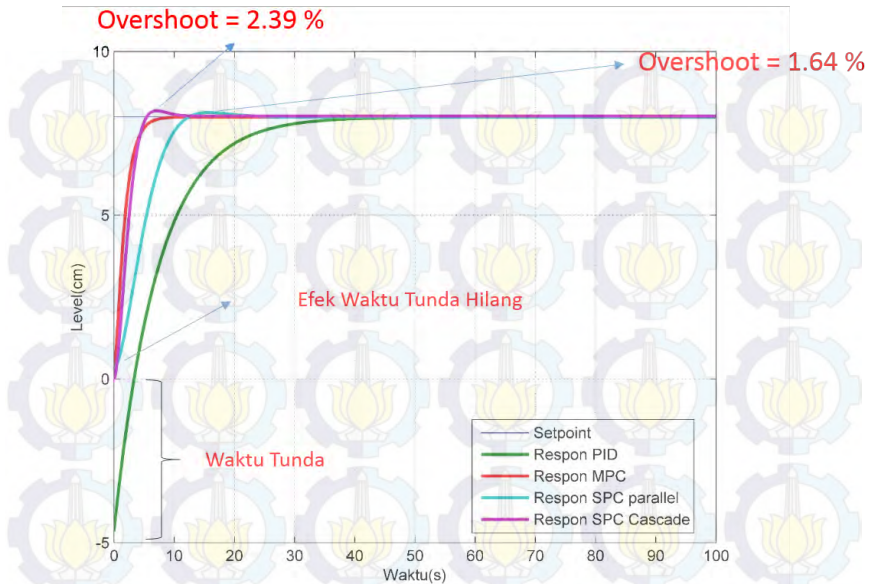


Gambar 4. 13. Respon sistem dengan kontroler SPC konfigurasi *parallel* dengan variasi N_p

4.5. Perbandingan Kontroler PI dan Prediktif

4.5.1. Perbandingan Kontroler PI dan Prediktif Saat Uji Coba *Set Point Sinyal Step*

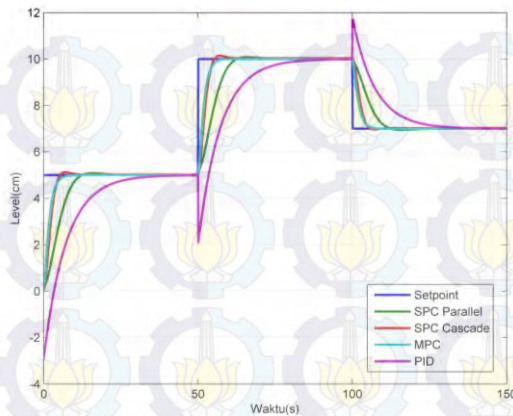
Pada simulasi ini dilakukan perbandingan antara keempat kontroler yang digunakan. Model yang digunakan pada perbandingan ini adalah model hasil penurunan matematis dengan pendekatan waktu tunda menggunakan pendekatan Taylor orde pertama. Dari hasil yang didapat pada Gambar 4. 14, kontroler prediktif dan kontroler PID dapat menghilangkan waktu tunda yang terjadi. Kontroler SPC konfigurasi cascade memiliki respon yang paling cepat dengan nilai waktu konstan sebesar 2.207 detik, namun memiliki overshoot sebesar 1.30 %. Kontroler SPC konfigurasi parallel juga memiliki *overshoot* sebesar 3.95 %, sedangkan kontroler MPC dan PI tidak memiliki *overshoot* tapi memiliki waktu konstan yang lama.



Gambar 4. 14. Perbanding respon dengan masukan sinyal *step*

4.5.2. Perbandingan Kontroler PI dan Prediktif Saat Uji Coba *Set Point Tracking*

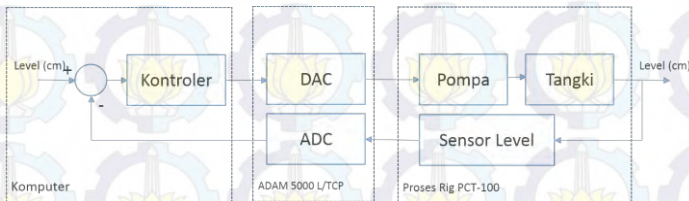
Pada simulasi ini dilakukan perbandingan antara empat kontroler yang digunakan terhadap perubahan set point dan didapatkan hasil seperti pada Gambar 4. 15. Dari hasil yang didapat SPC konfigurasi *cascade* memiliki respon yang paling baik dengan waktu konstan sebesar 3.3 detik dan *overshoot* sebesar 2.39 %. Kontroler PI tidak mampu menghilangkan efek dari waktu tunda sehingga pada saat terjadi perubahan pada nilai *set point* respon sinyal keluaran memiliki lonjakan nilai yang besar. Hal ini dikarenakan model pendekatan yang digunakan adalah deret Taylor orde pertama. Pada pengujian ini terlihat bahwa hasil pengujian kontroler prediktif mampu menghilangkan efek dari waktu tunda sehingga tujuan dari penelitian ini tercapai. Perancangan kontroler SPC konfigurasi *cascade* sangat cocok digunakan pada pengujian variasi nilai set point ini, karena tujuan dari kontroler ini memang digunakan untuk memudahkan operator dalam merubah *set point* sistem.



Gambar 4. 15. Perbandingan respon sistem dengan kontroler yang berbeda

4.6. Implementasi

Pada bagian ini, kontroler PI dan MPC *regulatory* digunakan pada plant PCT-100 dengan kondisi variasi set point. Sementara kontroler SPC konfigurasi *cascade* dan *parallel* tidak dilakukan implementasi. Pada Gambar 4. 16 terlihat skema implementasi yang dilakukan pada penelitian ini. Implementasi akan dilakukan dengan menggunakan perangkat akuisisi data yang menghubungkan kontroler dengan *plant*. Data analog dari sensor akan dikeluarkan melalui *control module* PCT-100 dan dikirimkan ke komputer yang berisi program kontroler melalui perangkat akuisisi data.



Gambar 4. 16. Skema implementasi

Pada Gambar 4. 17 terlihat *hardware* yang digunakan saat implementasi. Perangkat akuisisi data yang digunakan memiliki bagian

analog to digital conversion (ADC) dan *digital to analog conversion* (DAC) yang digunakan untuk merubah data pembacaan data dari *plant* dan dikirimkan ke kontroler maupun sebaliknya.

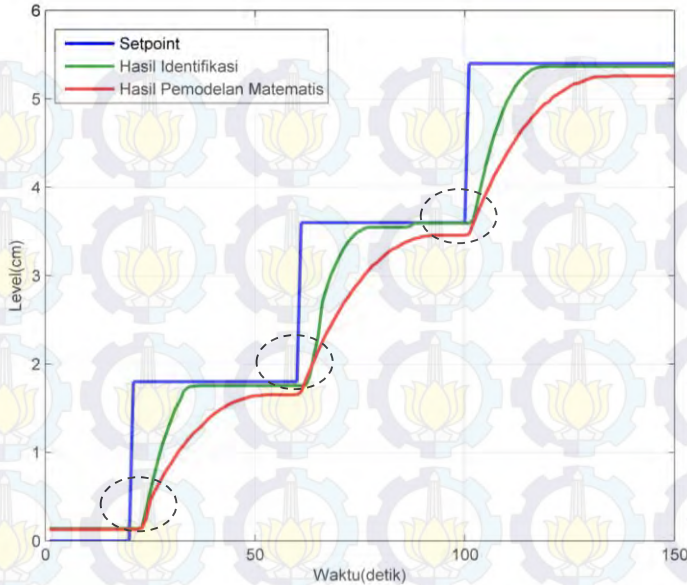


Gambar 4. 17. Susunan skema implementasi berdasarkan *hardware*

4.6.1. Kontroler PI

Pada implementasi yang dilakukan adalah kontroler yang digunakan diprogram pada komputer. Waktu sampling yang digunakan saat implementasi adalah sebesar 1000ms. Simulasi ini dilakukan dengan variasi *set point* sebesar 1.8cm, 3.6cm, dan 5.4cm. Nilai tersebut didapat dengan persamaan antara ketinggian level air dengan tegangan yang diberikan, dimana untuk ketinggian maksimum tangki sebesar 18 cm diberi masukan dengan tegangan maksimal 10 V. Didapat hubungan antara rentang tegangan dan ketinggian air yaitu 1.8V/cm. Pada Gambar 4. 18 terlihat grafik hasil implementasi kontroler PI. Saat diimplementasikan kontroler PI dengan pemodelan hasil model matematis memiliki respon yang sangat lama. Hal tersebut disebabkan oleh ppengkondisian sinyal yang diabaikan dalam penurunan model matematis *plant*. Sementara respon sistem dengan pemodelan hasil identifikasi mempunyai respon dengan waktu konstan sebesar 4.95 detik dan mendapatkan efek dari waktu tunda sebesar 3.29 detik. Saat pengujian ini, kontroler PI dengan model hasil pemodelan matematis belum mencapai kondisi waktu tunak.

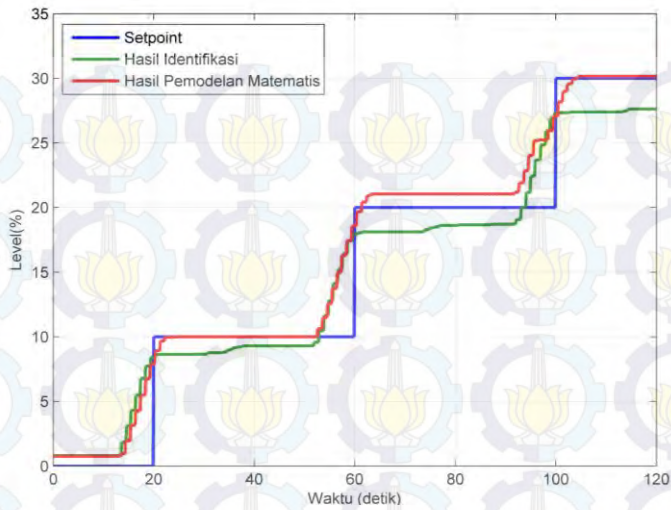
Pada saat diberi kontroler PI, pompa pada waktu tertentu mengalami keadaan *bouncing* dimana keadaan tersebut membuat *contact* pada relay yang terdapat pada pompa akan beroperasi nyala dan mati. Kondisi ini disebabkan parameter kontroler yang digunakan tidak sesuai dengan kondisi *plant*. Hal ini dapat menyebabkan *lifetime* dari peralatan yang digunakan berkurang.



Gambar 4. 18. Respon implementasi kontroler PI

4.6.2. Kontroler MPC

Pada implementasi kontroler MPC yang dilakukan, informasi *set point* sudah diberikan pada kondisi awal. Pada Gambar 4. 19 terlihat respon kontroler MPC dengan variasi *set point* menggunakan dua model berbeda yaitu hasil identifikasi dan penurunan model matematis. Terlihat bahwa respon dapat memprediksi sinyal keluaran sebelum nilai *set point* berubah. Dengan menggunakan nilai N_p sebesar 100, terlihat pada respon bahwa sinyal keluaran diprediksi 10 detik sebelum terjadi perubahan *set point*. Pada kasus implementasi ini, kondisi tangki pada pembacaan nol dimulai dengan ketinggian sebesar 0.78 cm. Hal ini diakibatkan pembacaan sensor yang tidak presisi. Dalam keadaan ini, harus dilakukan kalibrasi ulang pada sensor. Pada implementasi ini, sinyal kontrol yang diberikan ke pompa dibatasi sesuai dengan rentang kerja pompa yaitu 0-10 V.



Gambar 4. 19. Kontroler MPC saat implementasi



BAB 5

PENUTUP

5.1. Kesimpulan

Dari hasil simulasi dan implementasi yang dilakukan, dapat diambil kesimpulan-kesimpulan penting antara lain adalah:

- Kontroler prediktif yang digunakan dapat menghilangkan efek dari waktu tunda.
- Pengujian kontroler dengan model hasil identifikasi dan penurunan berdasarkan hukum fisis mendapatkan hasil yang berbeda diakibatkan beberapa parameter sistem yang diabaikan saat penurunan matematis.
- Perancangan kontroler MPC sangat bergantung pada pemodelan *plant*.
- Saat pengujian variasi nilai *set point*, SPC konfigurasi *cascade* mendapatkan hasil yang paling bagus yaitu memiliki konstanta waktu paling kecil sebesar 1.026 detik dengan model hasil identifikasi.
- Saat implementasi kontroler MPC, pemodelan hasil penurunan rumus matematika mendapatkan respon yang paling baik dengan kesalahan waktu tunak sebesar 0%.

5.2. Saran

Untuk pengembangan penelitian selanjutnya, penulis menyarankan beberapa hal yaitu:

- Pemodelan waktu tunda menggunakan pendekatan *pade* atau Mc'Laurin.
- Mempertimbangkan pengkondisian sinyal dalam penurunan model matematis sistem.
- Untuk mendapatkan model yang tepat, dilakukan pemodelan dengan model yang lain seperti ARX, ARMA, atau ARMAX.
- Penerapan kontroler prediktif diimplementasikan pada *plant* yang ada di industri

Halaman ini sengaja dikosongkan

LAMPIRAN

Penurunan rumus kontroler SPC

Sesuai dengan matrik yang telah didapatkan pada Persamaan (3-25) maka dapat dicari matrik Lzerot dengan rumus seperti pada Persamaan (6-1) dan didapatkan nilai seperti pada Persamaan (6-2).

$$L(0)^T = \sqrt{1-a^2} \begin{bmatrix} 1 & -a & a^2 & -a^3 & \dots & (-1)^{N-1} a^{N-1} \end{bmatrix} \quad (6-1)$$

$$L(0)^T = \begin{bmatrix} 0.8660 & -0.4330 & 0.2165 & -0.1083 & 0.0541 \end{bmatrix} \quad (6-2)$$

Setelah mendapatkan matrik Lzerot maka akan dicari matrik *augmented* dengan memasukkan pada rumus sesuai dengan Persamaan (2-49) – (2-51) dan didapatkan nilai seperti pada Persamaan (6-3) dan (6-4).

$$A_{-e} = \begin{bmatrix} 0.9901 & 0.0015 & 0 & 0 \\ -0.0054 & 1 & 0 & 0 \\ 0.0537 & 0.0001 & 1 & 0 \\ -0.0896 & 0.0153 & 0 & 1 \end{bmatrix} \quad (6-3)$$

$$B_{-e} = \begin{bmatrix} 0.1668 \\ 0.1 \\ 0.0090 \\ -0.0135 \end{bmatrix}$$

$$C_{-e} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6-4)$$

Matrik pembobot pada *state* (Q) dan pada masukan (R) didapatkan sesuai dengan persamaan (6-5).

$$Q = (C_{-e})^T \times C_{-e} \quad (6-5)$$

$$R = 0.01$$

Selanjutnya dihitung matrik Ω dan Ψ untuk mencari nilai Δu sesuai dengan Persamaan (6-6)

$$\Delta u = -L(0)^T \Omega^{-1} \Psi_X(k) \quad (6-6)$$

Maka untuk mendapatkan matrik Ω dicari terlebih dahulu matrik $\phi(m)^T$ seperti pada Persamaan (6-7).

$$\phi(m)^T = \sum_{i=0}^{m-1} A^{m-i-1} BL(i)^T = S_c(m) \quad (6-7)$$

Karena $N_p = 5$, maka nilai $m=5$ dan Persamaan (6-7) dapat diuraikan sesuai dengan Persamaan (6-8)

$$\begin{aligned} S_c(1) &= BL(0)^T \\ S_c(2) &= AS_c(1) + S_c(1)A_l^T \\ S_c(m) &= AS_c(m-1) + S_c(1)(A_l^{m-1})^T \end{aligned} \quad (6-8)$$

Sesuai dengan fungsi Laguerre nilai A_l dapat ditulis sesuai dengan Persamaan (6-9). Maka didapatkan nilai $S_c(m)$ seperti pada Persamaan (6-10).

$$\begin{aligned} A_l &= \begin{bmatrix} a & 0 & 0 & 0 & 0 \\ 1-a^2 & a & 0 & 0 & 0 \\ -a(1-a^2) & 1-a^2 & a & 0 & 0 \\ a^2(1-a^2) & -a(1-a^2) & 1-a^2 & a & 0 \\ -a^3(1-a^2) & a^2(1-a^2) & -a(1-a^2) & 1-a^2 & a \end{bmatrix} \\ &= \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0.75 & 0.5 & 0 & 0 & 0 \\ -0.375 & 0.75 & 0.5 & 0 & 0 \\ 0.1875 & -0.375 & 0.75 & 0.5 & 0 \\ -0.09375 & 0.1875 & -0.375 & 0.75 & 0.5 \end{bmatrix} \end{aligned} \quad (6-9)$$

$$S_c(1) = \begin{bmatrix} 0.1668 \\ 0.1 \\ 0.0090 \\ -0.0135 \end{bmatrix} \times \begin{bmatrix} 0.8660 & -0.04330 & 0.2165 & -0.1083 & 0.0541 \end{bmatrix}$$

$$= \begin{bmatrix} 0.1445 & -0.0722 & 0.0361 & -0.0181 & 0.0090 \\ 0.0866 & -0.0433 & 0.0217 & -0.0108 & 0.0054 \\ 0.0078 & -0.0039 & 0.0020 & -0.0010 & 0.0005 \\ -0.0117 & 0.0059 & -0.0029 & 0.0015 & -0.0007 \end{bmatrix} \quad (6-10)$$

Dengan melakukan perulangan sebanyak lima kali maka didapatkan matrik $\phi(m)^T$ seperti pada Persamaan (6-11).

$$\phi(5)^T = \begin{bmatrix} 0.2720 & 0.2150 & 0.0439 & -0.0755 & 0.0086 \\ 0.1631 & 0.1262 & 0.0263 & -0.0453 & 0.0051 \\ 0.0622 & 0.0213 & -0.0039 & -0.0046 & 0.0031 \\ -0.0931 & -0.0318 & 0.0059 & 0.0069 & -0.0047 \end{bmatrix} \quad (6-11)$$

Setelah memperoleh matrik $\phi(m)^T$ maka dicari matrik Ω dan Ψ sesuai dengan Persamaan (6-12) dan (6-13).

$$\Omega = \left(\sum_{m=1}^{N_p} \phi(m)Q\phi(m)^T + R_L \right) \quad (6-12)$$

$$\Psi = \left(\sum_{m=1}^{N_p} \phi(m)QA^m \right) \quad (6-13)$$

Dan didapatkan nilai seperti pada Persamaan (6-14) dan (6-15).

$$\Omega = \begin{bmatrix} 10.0124 & 0.0028 & -0.0012 & -0.0003 & 0.0005 \\ 0.0028 & 10.0009 & -0.0003 & -0.0002 & 0.0002 \\ -0.0012 & -0.0003 & 10.0001 & 0 & 0 \\ -0.0003 & -0.0002 & 0 & 10.0001 & 0 \\ 0.0005 & 0.0002 & 0 & 0 & 10 \end{bmatrix} \quad (6-14)$$

$$\Psi = \begin{bmatrix} 0.0598 & -0.0072 & 0.0850 & -0.1272 \\ 0.0126 & -0.0015 & 0.0122 & -0.0182 \\ -0.0055 & 0.0007 & -0.0071 & 0.0106 \\ -0.0014 & 0.0002 & -0.0009 & 0.0013 \\ 0.0021 & -0.0003 & 0.0022 & -0.0033 \end{bmatrix} \quad (6-15)$$

Selanjutnya akan dilakukan perhitungan nilai Δu dan dilakukan inisialisasi terhadap parameter yang akan dihitung seperti pada Persamaan (6-16).

$$\begin{aligned} y &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ u &= 0 \\ x_m &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ r &= [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1] \\ x(k) = [x_m; y - r(k)] &= \begin{bmatrix} 0 \\ 0 \\ -1 \\ -1 \end{bmatrix} \end{aligned} \quad (6-16)$$

Dari hasil simulasi yang dilakukan akan ditampilkan lima nilai pertama dari Δu sesuai dengan Persamaan (6-17) dan u seperti Persamaan (6-18).

$$\Delta u = [1.1330 \quad 1.4346 \quad 0.8188 \quad 0.9140 \quad 0.9696] \quad (6-17)$$

$$u = [0.0218 \quad 0.0218 \quad 0.0217 \quad 0.0 \quad 5.2700] \quad (6-18)$$

Dihitung nilai keluaran dan *state* sesuai dengan Persamaan (6-19) dan (6-20).

$$x_m = A_p x_m + B_p u \quad (6-19)$$

$$y = C_p x_m \quad (6-20)$$

Listing program

1. Kontroler PI

```
%% Program Simulasi Kontroler PI %%
Clear all
Clc
%% Plant
P=tf([1.76],[108.57 1],'InputDelay',4.23);
%model penurunan matematis
P1=tf([9.2],[2636.19 350.65
1],'InputDelay',2.4697); %model hasil
identifikasi
%% State space Model
[Ap,Bp,Cp,Dp]=tf2ss([-22.72 9.2],[2636.19 350.65
1]) %model hasil identifikasi
[Ap1,Bp1,Cp1,Dp1]=tf2ss([-7.44 1.76],[108.57 1])
%model penurunan rumus
[Ad,Bd,Cd,Dd]=c2dm(Ap,Bp,Cp,Dp,0.1)
[Ad1,Bd1,Cd1,Dd1]=c2dm(Ap1,Bp1,Cp1,Dp1,0.1)
%% Controller
ti=31.66; %hasil identifikasi
Kp=6.66; %hasil identifikasi
ti1=110.685; %penurunan rumus
Kp1=2.896; %penurunan rumus
C=tf([Kp*ti Kp],[ti 0]);
C1=tf([Kp1*ti1 Kp1],[ti1 0]);
[Acc,Bcc,Ccc,Dcc]=tf2ss([Kp*ti Kp],[ti 0]);
[Acc1,Bcc1,Ccc1,Dcc1]=tf2ss([Kp1*ti1 Kp1],[ti1
0]);
[Ac,Bc,Cc,Dc]=c2dm(Acc,Bcc,Ccc,Dcc,0.1)
```

```

[Ac1,Bc1,Cc1,Dc1]=c2dm(Acc1,Bcc1,Ccc1,Dcc1,0.1)
%% Inisialisasi
r1=8*ones(1,5001);
In1=5;
In11=2; %masukan 1 untuk sinyal
referensi/gangguan
In2=3; %masukan 2 untuk sinyal
referensi/gangguan
In3=7; %masukan 3 untuk sinyal
referensi/gangguan
m=0;
for i = 1:10002;
    w(i)=5;
end
for i = 1:7501;
    q(i)=1;
end
t=0:0.1:250;
t11=250:0.1:1000;
t1=0:0.1:1000.1;
t2=0:0.1:500;
t3=0:0.1:600;
%% Set point Tracking
for k = 1:3
    for k=1;
        r3(1,1:2001)=5*ones(1,2001);
    end
    for k=2;
        r3(1,2002:4001)=10*ones(1,2000);
    end
    for k=3;
        r3(1,4002:6001)=7*ones(1,2000);
    end
end
%% Transfer Function
T=feedback(C*P,1) %Set point
T1 = feedback(P,C) %Disturbance
T2=feedback(C1*P1,1) %Set point
T3 = feedback(P1,C1) %Disturbance
%% Disturbance Hasil Pemodelan Matematis

```

```

for k = 1:2
    m=m+1;
    for k = 1;
        B1(1:2501,1) = (In1*step(T,t));
        B1(1:2501,2) = (In1*step(T,t));
        B1(1:2501,3) = (In1*step(T,t));
    end
    for k=2;
        B1(2502:10002,1)=(In1*step(T,t11))-
        (In11*lsim(T1,q,t11));
        B1(2502:10002,2)=(In1*step(T,t11))-
        (In2*lsim(T1,q,t11));
        B1(2502:10002,3)=(In1*step(T,t11))-
        (In3*lsim(T1,q,t11));
    end
end
%% Disturbance Hasil Identifikasi
for k = 1:2
    m=m+1;
    for k = 1;
        B2(1:2501,1) = (In1*step(T2,t));
        B2(1:2501,2) = (In1*step(T2,t));
        B2(1:2501,3) = (In1*step(T2,t));
    end
    for k=2;
        B2(2502:10002,1)=(In1*step(T2,t11))-
        (In11*lsim(T3,q,t11));
        B2(2502:10002,2)=(In1*step(T2,t11))-
        (In2*lsim(T3,q,t11));
        B2(2502:10002,3)=(In1*step(T2,t11))-
        (In3*lsim(T3,q,t11));
    end
end
end

```

2. Kontroler MPC Regulator

```

%% Program Simulasi Kontroler MPC pada level
regulatory %%
clear all

```



```

clc
%% masukkan state space continous hasil
pemodelan orde 1
%Ad=[-0.0092];           %Penurunan Model Matematis
%Bd=[1];                 %Penurunan Model Matematis
%Cd=[0.0168];           %Penurunan Model Matematis
%Dd=[-0.0685];          %Penurunan Model Matematis
Ad=[-0.1330 -0.0004; 1 0]; %Hasil Identifikasi
Bd=[1;0];               %Hasil Identifikasi
Cd=[-0.0086 0.0035];    %Hasil Identifikasi
Dd=[0];                 %Hasil Identifikasi
Np=100;
Np1=500;
Np2=1000;
Nc=5;
Delta_t=0.1;
[Ap,Bp,Cp,Dp]=c2dm(Ad,Bd,Cd,Dd,Delta_t);
%% Mencari matrik augmented Hasil Pemodelan
Matematis
[m1,n1]=size(Cp);
[n1,n_in]=size(Bp);
A_e=eye(n1+m1,n1+m1);
A_e(1:n1,1:n1)=Ap;
A_e(n1+1:n1+m1,1:n1)=Cp*Ap;
B_e=zeros(n1+m1,n_in);
B_e(1:n1,:)=Bp;
B_e(n1+1:n1+m1,:)=Cp*Bp;
C_e=zeros(m1,n1+m1);
C_e(:,n1+1:n1+m1)=eye(m1,m1);
%% mencari fungsi F dan Phi untuk Np=100
n=n1+m1;
h(1,:)=C_e;
F(1,:)=C_e*A_e;
for kk=2:Np
    h(kk,:)=h(kk-1,:)*A_e;
    F(kk,:)=F(kk-1,:)*A_e;
end
v=h*B_e;
Phi=zeros(Np,Nc);
Phi(:,1)=v;

```

```

for i=2:Nc
    Phi(:,i)=[zeros(i-1,1);v(1:Np-i+1,1)];
end
BarRs=ones(Np,1);
Phi_Phi=Phi'*Phi;
Phi_F=Phi'*F;
Phi_R=Phi'*BarRs;
%% mencari fungsi F dan Phi untuk Np=500
n=n1+m1;
h1(1,:)=C_e;
F1(1,:)=C_e*A_e;
for kk=2:Np1
    h1(kk,:)=h1(kk-1,:)*A_e;
    F1(kk,:)=F1(kk-1,:)*A_e;
end
v1=h1*B_e;
Phi1=zeros(Np1,Nc);
Phi1(:,1)=v1;
for i=2:Nc
    Phi1(:,i)=[zeros(i-1,1);v1(1:Np1-i+1,1)];
end
BarRs1=ones(Np1,1);
Phi_Phi1=Phi1'*Phi1;
Phi_F1=Phi1'*F1;
Phi_R1=Phi1'*BarRs1;
%% mencari fungsi F dan Phi untuk Np=1000
n=n1+m1;
h2(1,:)=C_e;
F2(1,:)=C_e*A_e;
for kk=2:Np2
    h2(kk,:)=h2(kk-1,:)*A_e;
    F2(kk,:)=F2(kk-1,:)*A_e;
end
v2=h2*B_e;
Phi2=zeros(Np2,Nc);
Phi2(:,1)=v2;
for i=2:Nc
    Phi2(:,i)=[zeros(i-1,1);v2(1:Np2-i+1,1)];
end
BarRs2=ones(Np2,1);

```

```

Phi_Phi2=Phi2'*Phi2;
Phi_F2=Phi2'*F2;
Phi_R2=Phi2'*BarRs2;
%% Inisialisasi
[n,n_in]=size(B_e);
%xm=[0];xm1=[0];xm2=[0];xm3=[0];xm4=[0];
%xm5=[0];xm6=[0];xm7=[0];xm8=[0];
xm=[0;0];xm1=[0;0];xm2=[0;0];xm3=[0;0];xm4=[0;0];
;
xm5=[0;0];xm6=[0;0];xm7=[0;0];xm8=[0;0];
Xf=zeros(n,1);
Xf1=zeros(n,1);
Xf2=zeros(n,1);
N_sim=5000;
r1=8*ones(1,N_sim+1);
u=0; %u(k-1)=0
y=0;y21=0;y31=0;y4=0;y_11=0;y_111=0;y_9=0;y_10=0
;y_12=0;y_4=0;
B1=[1];
%% Respon Fungsi Step Np=100
for kk=1:5000;

DeltaU1=inv(Phi_Phi+100*eye(Nc,Nc))*(Phi_R*r1(kk)
)-Phi_F*Xf);
    deltau1=DeltaU1(1,1);
    u=u+deltau1;
    u123(kk)=u;
    y1(kk)=y;
    xm_old=xm;
    xm=Ap*xm+Bp*u;
    y=Cp*xm;
    deltau11(kk)=deltau1;
    Xf=[xm-xm_old;y];
end
%% Respon Fungsi Step Np=500
for kk=1:5000;

DeltaU_1=inv(Phi_Phi1+1*eye(Nc,Nc))*(Phi_R1*r1(k
k)-Phi_F1*Xf);
    deltau_1=DeltaU_1(1,1);

```

```

        u=u+deltau_1;
        u1(kk)=u;
        y11(kk)=y_11;
        xm_old4=xm4;
        xm4=Ap*xm4+Bp*u;
        y_11=Cp*xm4;
        deltau_11(kk)=deltau_1;
        Xf=[xm4-xm_old4;y_11];
    end
    %% Respon Fungsi Step Np=1000
    for kk=1:5000;

        DeltaU_11=inv(Phi_Phi2+1*eye(Nc,Nc))*(Phi_R2*r1(
            kk)-Phi_F2*Xf);
        deltau_11=DeltaU_11(1,1);
        u=u+deltau_11;
        u1(kk)=u;
        y111(kk)=y_111;
        xm_old5=xm5;
        xm5=Ap*xm5+Bp*u;
        y_111=Cp*xm5;
        deltau_111(kk)=deltau_11;
        Xf=[xm5-xm_old5;y_111];
    end
    %% Disturbance
    for ss=1:300;

        DeltaU=inv(Phi_Phi+1*eye(Nc,Nc))*(Phi_R*r1(ss)-
            Phi_F*Xf);
        deltau=DeltaU(1,1);
        u=u+deltau;
        u4(ss)=u;
        y4(ss)=y_4;
        xm_old3=xm3;
        xm3=Ap*xm3+Bp*u;
        y_4=Cp*xm3;
        Xf=[xm3-xm_old3;y_4];
    end
    for ss=1:700;

```



```
DeltaU=inv(Phi_Phi+1*eye(Nc,Nc))*(Phi_R*r1(ss)-
Phi_F*Xf);
```

```
    deltau=DeltaU(1,1);
    u=u+deltau;
    u5(ss)=u;
    y5(ss)=y_4;
    wk=2;
    xm_old3=xm3;
    xm3=Ap*xm3+Bp*u-B1*wk;
    y_4=Cp*xm3;
    Xf=[xm3-xm_old3;y_4];
```

```
end
```

```
for ss1=1:700;
```

```
DeltaU=inv(Phi_Phi+1*eye(Nc,Nc))*(Phi_R*r1(ss1)-
Phi_F*Xf);
```

```
    deltau=DeltaU(1,1);
    u=u+deltau;
    u6(ss1)=u;
    y6(ss1)=y_4;
    wk=3;
    xm_old3=xm3;
    xm3=Ap*xm3+Bp*u-B1*wk;
    y_4=Cp*xm3;
    Xf=[xm3-xm_old3;y_4];
```

```
end
```

```
for ss2=1:700;
```

```
DeltaU=inv(Phi_Phi+1*eye(Nc,Nc))*(Phi_R*r1(ss2)-
Phi_F*Xf);
```

```
    deltau=DeltaU(1,1);
    u=u+deltau;
    u7(ss2)=u;
    y7(ss2)=y_4;
    wk=7;
    xm_old3=xm3;
    xm3=Ap*xm3+Bp*u-B1*wk;
    y_4=Cp*xm3;
    Xf=[xm3-xm_old3;y_4];
```

end

3. SPC konfigurasi cascade

```
%Program SPC Konfigurasi Cascade%
clear all
clc
%% Persamaan State plant dan kontroler
Ap=[0.9991]; %Penurunan Model Matematis
Bp1=[0.1]; %Penurunan Model Matematis
Bp2=[1]; %Penurunan Model Matematis
Cp=[0.0168]; %Penurunan Model Matematis
Dp=[-0.0685]; %Penurunan Model Matematis
%Ap=[-0.9868 0; 0.0993 1]; %Hasil Identifikasi
%Bp1=[0.0993;0.005]; %Hasil Identifikasi
%Bp2=[1;0]; %Hasil Identifikasi
%Cp=[-0.0086 0.0035]; %Hasil Identifikasi
%Dp=[0]; %Hasil Identifikasi
%Ac=[1]; %Hasil Identifikasi
%Bc=[0.1]; %Hasil Identifikasi
%Cc=[0.2104]; %Hasil Identifikasi
%Dc=[6.66]; %Hasil Identifikasi
Ac=[1]; %Penurunan Model Matematis
Bc=[0.1]; %Penurunan Model Matematis
Cc=[0.0262]; %Penurunan Model Matematis
Dc=[2.896]; %Penurunan Model Matematis
Dp1=[1];
Dp2=[1];
I=eye(1);
M=inv(I+Dc*Dp1);
%% Konfigurasi Cascade
A=[Ap-(Bp1*M*Dc*Cp) Bp1*M*Cc; (Bc*Dp1*M*Dc*Cp) -
(Bc*Cp) Ac-(Bc*Dp1*M*Cc)]
B1=[Bp1*M*Dc;Bc-(Bc*Dp1*M*Dc)]
B2=[Bp2-(Bp1*M*Dc*Dp2) Bp1*M; (Bc*Dp1*M*Dc*Dp2) -
(Bc*Dp2) -Bc*Dp1*M];
C=[Cp-(Dp1*M*Dc*Cp) Dp1*M*Cc;-M*Dc*Cp M*Cc]
D1=[Dp1*M*Dc; M*Dc]
D2=[Dp2-(Dp1*M*Dc*Dp2) Dp1*M; -M*Dc*Dp2 M];
```

```

N_sim=3000;
[m1,n1]=size(C);
[n1,n_in]=size(B1);
%% Inisialisasi Parameter Fungsi Laguerre
a1=0.5;
N1=5;
a=[a1];
N=[N1];
Np=1000;
Np=100;Np1=500;Np2=1000;
%% Mendapatkan Matrik Lzerot
N_pa=sum(N);
M=zeros(n_in,N_pa);
M_dul=zeros(n_in,N_pa);
k0=1;
[A1,L0]=lagd(a(k0),N(k0));
M_dul(1,1:N(1))=L0';
cc=N(1);
for k0=2:n_in;
    [A1,L0]=lagd(a(k0),N(k0));
    M_dul(k0,cc+1:cc+N(k0))=L0';
    cc=cc+N(k0);
end
Lzerot=M_dul;
%% Mendapatkan matrik augmentasi dan Q,R
A_e=eye(n1+m1,n1+m1);
A_e(1:n1,1:n1)=A;
A_e(n1+1:n1+m1,1:n1)=C*A;
B_e=zeros(n1+m1,n_in);
B_e(1:n1,:)=B1;
B_e(n1+1:n1+m1,:)=C*B1;
C_e(:,n1+1:n1+m1)=eye(m1,m1);
Q=0.5*C_e'*C_e;Q1=Q;Q2=Q;
%% Mencari Matrik E dan H
[n,n_in]=size(B_e);
N_pa=sum(N);
E=zeros(N_pa,N_pa);
H=zeros(N_pa,n);
R=0.001*eye(N_pa,N_pa);R1=R;R2=R;
S_in=zeros(n,N_pa);

```

```

[A1,L0]=lagd(a(1),N(1));
S_in(:,1:N(1))=B_e(:,1)*L0';
In_s=1;
for jj=2:n_in;
    [A1,L0]=lagd(a(jj),N(jj));
    In_s=N(jj-1)+In_s;
    In_e=In_s+N(jj)-1;
    S_in(:,In_s:In_e)=B_e(:,jj)*L0';
end
%% Mencari Matrik E dan H Np=100
S_sum=S_in;
phi=S_in;
E=(phi)'*Q*(phi);
H=phi'*Q*A_e;
for i=2:Np;
    Eae=A_e^i;
    [A1,L0]=lagd(a(1),N(1));
    S_sum(:,1:N(1))=A_e*S_sum(:,1:N(1))+
    S_in(:,1:N(1))*(A1^(i-1))';
    In_s=1;
    for kk=2:n_in;
        [A1,L0]=lagd(a(kk),N(kk));
        In_s=N(kk-1)+In_s;
        In_e=In_s+N(kk)-1;

    S_sum(:,In_s:In_e)=A_e*S_sum(:,In_s:In_e)+
    S_in(:,In_s:In_e)*(A1^(i-1))';
    end
    phi=S_sum;
    E=E+phi'*Q*phi;
    H=H+phi'*Q*Eae;
end
E=E+R;
%% Mencari Matrik E dan H Np=500
S_sum1=S_in;
phi1=S_in;
E1=(phi1)'*Q1*(phi1);
H1=phi1'*Q1*A_e;
for i=2:Np1;
    Eae1=A_e^i;

```



```

    [A1,L0]=lagd(a(1),N(1));
    S_sum1(:,1:N(1))=A_e*S_sum1(:,1:N(1))+
    S_in(:,1:N(1))*(A1^(i-1))';
    In_s=1;
    for kk=2:n_in;
        [A1,L0]=lagd(a(kk),N(kk));
        In_s=N(kk-1)+In_s;
        In_e=In_s+N(kk)-1;

    S_sum1(:,In_s:In_e)=A_e*S_sum1(:,In_s:In_e)+
    S_in(:,In_s:In_e)*(A1^(i-1))';
    end
    phi1=S_sum1;
    E1=E1+phi1'*Q1*phi1;
    H1=H1+phi1'*Q1*Eae1;
end
E1=E1+R1;
%% Mencari Matrik E dan H Np=1000
S_sum2=S_in;
phi2=S_in;
E2=(phi2)'*Q2*(phi2);
H2=phi2'*Q2*A_e;
for i=2:Np;
    Eae2=A_e^i;
    [A1,L0]=lagd(a(1),N(1));
    S_sum2(:,1:N(1))=A_e*S_sum2(:,1:N(1))+
    S_in(:,1:N(1))*(A1^(i-1))';
    In_s=1;
    for kk=2:n_in;
        [A1,L0]=lagd(a(kk),N(kk));
        In_s=N(kk-1)+In_s;
        In_e=In_s+N(kk)-1;

    S_sum2(:,In_s:In_e)=A_e*S_sum2(:,In_s:In_e)+
    S_in(:,In_s:In_e)*(A1^(i-1))';
    end
    phi2=S_sum2;
    E2=E2+phi2'*Q2*phi2;
    H2=H2+phi2'*Q2*Eae2;
end
end

```

```
E2=E2+R2;
```

```
% Inisialisasi nilai vektor state, input, dan
output
y=zeros(m1,1);y1=y;y2=y;y3=y;y4=y;y5=y;y6=y;y7=y
;y8=y;
u=zeros(n_in,1);
xm=zeros(n1,1);xm1=xm;xm2=xm;xm3=xm;xm4=xm;xm5=x
m;xm6=xm;xm7=xm;xm8=xm;
r1=ones(1,N_sim+1);
r=zeros(1,N_sim+1);
sp=[r1;r];
%% Mendapatkan nilai output Np=100
Xf=[xm;y-(sp(:,1))];
for kk=1:N_sim;
    eta=-(E\H)*Xf;
    deltau=Lzerot*eta;
    u=u+deltau;
    deltau1(:,kk)=deltau;
    xm_old=xm;
    u_(1:n_in,kk)=u;
    y_(1:m1,kk)=y;
    xm_(1:n1,kk)=xm;
    xm=A*xm+B1*u;
    y=C*xm;
    Xf=[xm-xm_old;(y-sp(:,kk+1))];
end
%% Mendapatkan nilai output Np=500
Xf=[xm1;y1-(sp(:,1))];
for kk=1:N_sim;
    eta=-(E1\H1)*Xf;
    deltau=Lzerot*eta;
    u=u+deltau;
    deltau1(:,kk)=deltau;
    xm_old1=xm1;
    u_(1:n_in,kk)=u;
    y_(1:m1,kk)=y1;
    xm_(1:n1,kk)=xm1;
    xm1=A*xm1+B1*u;
```

```

        y1=C*xm1;
        Xf=[xm1-xm_old1; (y1-sp(:,kk+1))];
    end
    %% Mendapatkan nilai output Np=1000
    Xf=[xm2;y2-(sp(:,1))];
    for kk=1:N_sim;
        eta=-(E2\H2)*Xf;
        deltau=Lzerot*eta;
        u=u+deltau;
        deltau1(:,kk)=deltau;
        xm_old2=xm2;
        u_2(1:n_in, kk)=u;
        y_2(1:m1, kk)=y2;
        xm_2(1:n1, kk)=xm2;
        xm2=A*xm2+B1*u;
        y2=C*xm2;
        Xf=[xm2-xm_old2; (y2-sp(:,kk+1))];
    end
    %% Disturbance
    Xf=[xm5;(y5-sp(:,1))];
    for kk=1:300;
        eta=-(E\H)*Xf;
        deltau=Lzerot*eta;
        u=u+deltau;
        deltau1(:,kk)=deltau;
        y_5(1:m1, kk)=y5;
        xm_old5=xm5;
        u_5(1:n_in, kk)=u;
        xm_5(1:n1, kk)=xm5;
        xm5=A*xm5+B1*u;
        y5=C*xm5;
        Xf=[xm5-xm_old5; (y5-sp(:,kk+1))];
    end
    for kk=1:700;
        eta=-(E\H)*Xf;
        deltau=Lzerot*eta;
        wk=[2;2];
        u=u+deltau;
        deltau1(:,kk)=deltau;
        xm_old5=xm5;

```

```

u_5(1:n_in, kk)=u;
y_6(1:m1, kk)=y5;
xm_5(1:n1, kk)=xm5;
xm5=A*xm5+B1*u+B2*wk;
y5=C*xm5;
Xf=[xm5-xm_old5; (y5-sp(:, kk+1))];
end
for kk1=1:700;
eta=-(E\H)*Xf;
deltau=Lzerot*eta;
wk=[3;3];
u=u+deltau;
deltau1(:, kk1)=deltau;
xm_old5=xm5;
u_5(1:n_in, kk1)=u;
y_7(1:m1, kk1)=y5;
xm_5(1:n1, kk1)=xm5;
xm5=A*xm5+B1*u+B2*wk;
y5=C*xm5;
Xf=[xm5-xm_old5; (y5-sp(:, kk1+1))];
end
for kk2=1:700;
eta=-(E\H)*Xf;
deltau=Lzerot*eta;
wk=[7;7];
u=u+deltau;
deltau1(:, kk2)=deltau;
xm_old5=xm5;
u_5(1:n_in, kk2)=u;
y_8(1:m1, kk2)=y5;
xm_5(1:n1, kk2)=xm5;
xm5=A*xm5+B1*u+B2*wk;
y5=C*xm5;
Xf=[xm5-xm_old5; (y5-sp(:, kk2+1))];
end

```

4. SPC Konfigurasi parallel

```
%%Program SPC Konfigurasi Parallel%%
```



```

clear all
clc
%% Persamaan State plant dan kontroler
Ap=[0.9991]; %Penurunan Model Matematis
Bp1=[0.1]; %Penurunan Model Matematis
Bp2=[1]; %Penurunan Model Matematis
Cp=[0.0168]; %Penurunan Model Matematis
Dp=[-0.0685]; %Penurunan Model Matematis
%Ap=[-0.9868 0; 0.0993 1]; %Hasil Identifikasi
%Bp1=[0.0993;0.005]; %Hasil Identifikasi
%Bp2=[1;0]; %Hasil Identifikasi
%Cp=[-0.0086 0.0035]; %Hasil Identifikasi
%Dp=[0]; %Hasil Identifikasi
%Ac=[1]; %Hasil Identifikasi
%Bc=[0.1]; %Hasil Identifikasi
%Cc=[0.2104]; %Hasil Identifikasi
%Dc=[6.66]; %Hasil Identifikasi
Ac=[1]; %Penurunan Model Matematis
Bc=[0.1]; %Penurunan Model Matematis
Cc=[0.0262]; %Penurunan Model Matematis
Dc=[2.896]; %Penurunan Model Matematis
Dp1=[1];
Dp2=[1];
I=eye(1);
M=inv(I+Dc*Dp1);
%% Persamaan Matrik
A=[Ap-Bp1*M*Dc*Cp Bp1*M*Cc; (Bc*Dp1*M*Dc*Cp)-
(Bc*Cp) Ac-(Bc*Dp1*M*Cc)]
B1=[Bp1*M*Dc Bp1-(Bp1*M*Dc*Dp1);Bc-(Bc*Dp1*M*Dc)
(Bc*Dp1*M*Dc*Dp1)-Bc*Dp1]
B2=[Bp2-(Bp1*M*Dc*Dp2) Bp1*M; (Bc*Dp1*M*Dc*Dp2)-
(Bc*Dp2) -Bc*Dp1*M];
C=[Cp-(Dp1*M*Dc*Cp) Dp1*M*Cc; -M*Dc*Cp M*Cc]
D1=[Dp1*M*Dc Dp1-(Dp1*M*Dc*Dp1); M*Dc I-
M*Dc*Dp1]
D2=[Dp2-(Dp1*M*Dc*Dp2) Dp1*M; -M*Dc*Dp2 M];
N_sim=3000;
[m1,n1]=size(C);
[n1,n_in]=size(B1);
%% Inisialisasi Parameter Fungsi Laguerre

```

```

a1=0.5;
a2=0.5;
N1=10;
N2=10;
a=[a1 a2];
N=[N1 N2];
Np=100;Np1=500;Np2=1000;
%% Mendapatkan Matrik Lzerot
N_pa=sum(N);
M=zeros(n_in,N_pa);
M_dul=zeros(n_in,N_pa);
k0=1;
[A1,L0]=lagd(a(k0),N(k0));
M_dul(1,1:N(1))=L0';
cc=N(1);
for k0=2:n_in;
    [A1,L0]=lagd(a(k0),N(k0));
    M_dul(k0,cc+1:cc+N(k0))=L0';
    cc=cc+N(k0);
end
Lzerot=M_dul;
%% Mendapatkan matrik augmentasi dan Q,R
A_e=eye(n1+m1,n1+m1);
A_e(1:n1,1:n1)=A;
A_e(n1+1:n1+m1,1:n1)=C*A;
B_e=zeros(n1+m1,n_in);
B_e(1:n1,:)=B1;
B_e(n1+1:n1+m1,:)=C*B1;
C_e(:,n1+1:n1+m1)=eye(m1,m1);
Q=0.5*C_e'*C_e;Q1=Q;Q2=Q;
%% Mencari Matrik E dan H
[n,n_in]=size(B_e);
N_pa=sum(N);
E=zeros(N_pa,N_pa);
H=zeros(N_pa,n);
R=0.001*eye(N_pa,N_pa);R1=R;R2=R;
S_in=zeros(n,N_pa);
[A1,L0]=lagd(a(1),N(1));
S_in(:,1:N(1))=B_e(:,1)*L0';
In_s=1;

```

```

for jj=2:n_in;
    [A1,L0]=lagd(a(jj),N(jj));
    In_s=N(jj-1)+In_s;
    In_e=In_s+N(jj)-1;
    S_in(:,In_s:In_e)=B_e(:,jj)*L0';
end
%% Mencari Matrik E dan H Np=100
S_sum=S_in;
phi=S_in;
E=(phi)'*Q*(phi);
H=phi'*Q*A_e;
for i=2:Np;
    Eae=A_e^i;
    [A1,L0]=lagd(a(1),N(1));
    S_sum(:,1:N(1))=A_e*S_sum(:,1:N(1))+
    S_in(:,1:N(1))*(A1^(i-1))';
    In_s=1;
    for kk=2:n_in;
        [A1,L0]=lagd(a(kk),N(kk));
        In_s=N(kk-1)+In_s;
        In_e=In_s+N(kk)-1;

    S_sum(:,In_s:In_e)=A_e*S_sum(:,In_s:In_e)+
    S_in(:,In_s:In_e)*(A1^(i-1))';
    end
    phi=S_sum;
    E=E+phi'*Q*phi;
    H=H+phi'*Q*Eae;
end
E=E+R;
%% Mencari Matrik E dan H Np=500
S_sum1=S_in;
phi1=S_in;
E1=(phi1)'*Q1*(phi1);
H1=phi1'*Q1*A_e;
for i=2:Np1;
    Eae1=A_e^i;
    [A1,L0]=lagd(a(1),N(1));
    S_sum1(:,1:N(1))=A_e*S_sum1(:,1:N(1))+
    S_in(:,1:N(1))*(A1^(i-1))';

```

```

In_s=1;
for kk=2:n_in;
    [A1,L0]=lagd(a(kk),N(kk));
    In_s=N(kk-1)+In_s;
    In_e=In_s+N(kk)-1;

    S_sum1(:,In_s:In_e)=A_e*S_sum1(:,In_s:In_e)+
    S_in(:,In_s:In_e)*(A1^(i-1))';
    end
    phi1=S_sum1;
    E1=E1+phi1'*Q1*phi1;
    H1=H1+phi1'*Q1*Eae1;
end
E1=E1+R1;
%% Mencari Matrik E dan H Np=1000
S_sum2=S_in;
phi2=S_in;
E2=(phi2)'*Q2*(phi2);
H2=phi2'*Q2*A_e;
for i=2:Np;
    Eae2=A_e^i;
    [A1,L0]=lagd(a(1),N(1));
    S_sum2(:,1:N(1))=A_e*S_sum2(:,1:N(1))+
    S_in(:,1:N(1))*(A1^(i-1))';
    In_s=1;
    for kk=2:n_in;
        [A1,L0]=lagd(a(kk),N(kk));
        In_s=N(kk-1)+In_s;
        In_e=In_s+N(kk)-1;

        S_sum2(:,In_s:In_e)=A_e*S_sum2(:,In_s:In_e)+
        S_in(:,In_s:In_e)*(A1^(i-1))';
        end
        phi2=S_sum2;
        E2=E2+phi2'*Q2*phi2;
        H2=H2+phi2'*Q2*Eae2;
    end
    E2=E2+R2;
    %% Inisialisasi nilai vektor state, input, dan
    output

```



```

y=zeros(m1,1);y1=y;y2=y;y3=y;y4=y;y5=y;y6=y;y7=y
;y8=y;
u=zeros(n_in,1);
xm=zeros(n1,1);xm1=xm;xm2=xm;xm3=xm;xm4=xm;xm5=x
m;xm6=xm;xm7=xm;xm8=xm;
r1=8*ones(1,N_sim+1);
r=0.5*r1;
sp=[r1;r];
%% Mendapatkan nilai output Np=100
Xf=[xm;y-(sp(:,1))];
for kk=1:N_sim;
    eta=-(E\H)*Xf;
    deltau=Lzerot*eta;
    u=u+deltau;
    deltau1(:,kk)=deltau;
    xm_old=xm;
    u_1(1:n_in,kk)=u;
    y_1(1:m1,kk)=y;
    xm_1(1:n1,kk)=xm;
    xm=A*xm+B1*u;
    y=C*xm;
    Xf=[xm-xm_old;(y-sp(:,kk+1))];
end
%% Mendapatkan nilai output Np=500
Xf=[xm1;y1-(sp(:,1))];
for kk=1:N_sim;
    eta=-(E1\H1)*Xf;
    deltau=Lzerot*eta;
    u=u+deltau;
    deltau1(:,kk)=deltau;
    xm_old1=xm1;
    u_1(1:n_in,kk)=u;
    y_1(1:m1,kk)=y1;
    xm_1(1:n1,kk)=xm1;
    xm1=A*xm1+B1*u;
    y1=C*xm1;
    Xf=[xm1-xm_old1;(y1-sp(:,kk+1))];
end
%% Mendapatkan nilai output Np=1000
Xf=[xm2;y2-(sp(:,1))];

```

```

for kk=1:N_sim;
    eta=-(E2\H2)*Xf;
    deltau=Lzerot*eta;
    u=u+deltau;
    deltau1(:,kk)=deltau;
    xm_old2=xm2;
    u_2(1:n_in, kk)=u;
    y_2(1:m1, kk)=y2;
    xm_2(1:n1, kk)=xm2;
    xm2=A*xm2+B1*u;
    y2=C*xm2;
    Xf=[xm2-xm_old2; (y2-sp(:,kk+1))];
end
%% Disturbance
Xf=[xm5; (y5-sp(:,1))];
for kk=1:300;
    eta=-(E\H)*Xf;
    deltau=Lzerot*eta;
    u=u+deltau;
    deltau1(:,kk)=deltau;
    y_5(1:m1, kk)=y5;
    xm_old5=xm5;
    u_5(1:n_in, kk)=u;
    xm_5(1:n1, kk)=xm5;
    xm5=A*xm5+B1*u;
    y5=C*xm5;
    Xf=[xm5-xm_old5; (y5-sp(:,kk+1))];
end
for kk=1:700;
    eta=-(E\H)*Xf;
    deltau=Lzerot*eta;
    wk=[2;2];
    u=u+deltau;
    deltau1(:,kk)=deltau;
    xm_old5=xm5;
    u_5(1:n_in, kk)=u;
    y_6(1:m1, kk)=y5;
    xm_5(1:n1, kk)=xm5;
    xm5=A*xm5+B1*u-B2*wk;
    y5=C*xm5;

```

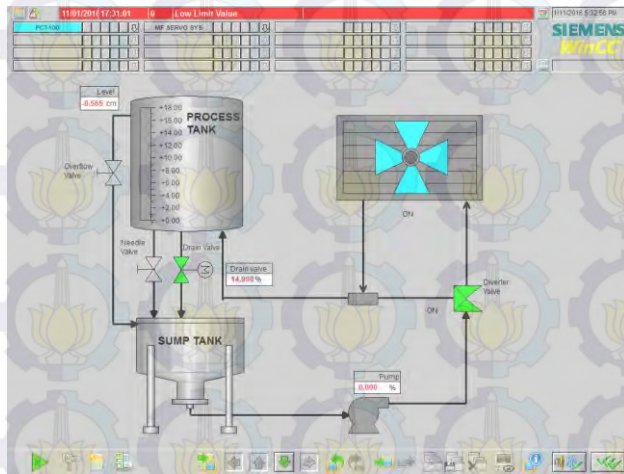
```

        Xf=[xm5-xm_old5; (y5-sp(:,kk1+1))];
    end
    for kk1=1:700;
        eta=-(E\H)*Xf;
        deltau=Lzerot*eta;
        wk=[3;3];
        u=u+deltau;
        deltau1(:,kk1)=deltau;
        xm_old5=xm5;
        u_5(1:n_in, kk1)=u;
        y_7(1:m1, kk1)=y5;
        xm_5(1:n1, kk1)=xm5;
        xm5=A*xm5+B1*u-B2*wk;
        y5=C*xm5;
        Xf=[xm5-xm_old5; (y5-sp(:,kk1+1))];
    end
    for kk2=1:700;
        eta=-(E\H)*Xf;
        deltau=Lzerot*eta;
        wk=[7;7];
        u=u+deltau;
        deltau1(:,kk2)=deltau;
        xm_old5=xm5;
        u_5(1:n_in, kk2)=u;
        y_8(1:m1, kk2)=y5;
        xm_5(1:n1, kk2)=xm5;
        xm5=A*xm5+B1*u-B2*wk;
        y5=C*xm5;
        Xf=[xm5-xm_old5; (y5-sp(:,kk2+1))];
    end
end

```

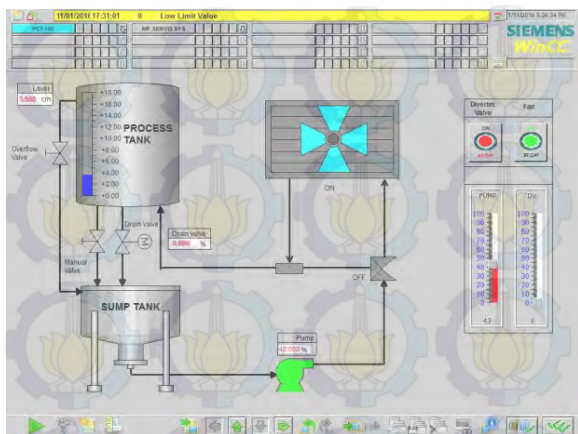
Tampilan HMI

1. Home screen



2. Controller setting screen





The screenshot displays the Siemens WinCC Graphics Designer interface. At the top, the title bar shows the file name 'High Limit Value' and the date/time '11/11/2016 7:45:15 PM'. The main window is titled 'Incoming alarm list'. It contains a table with the following columns: Date, Time, Priority, WinCC message text, Event, Status, Info, and Comment. The table has 31 rows, numbered 1 to 31. Rows 1 to 4 contain data for high and low limit value alarms. Row 1: Date 11/11/2016, Time 7:45:15, Priority 0, WinCC message text 'High Limit Value', Event 'High Limit Value', Status 'Acknowledge', Info 'Limit 0.1%', Comment 'High Limit Value'. Row 2: Date 11/11/2016, Time 7:45:15, Priority 0, WinCC message text 'High Limit Value', Event 'High Limit Value', Status 'Acknowledge', Info 'Limit 0.1%', Comment 'High Limit Value'. Row 3: Date 11/11/2016, Time 7:45:15, Priority 0, WinCC message text 'High Limit Value', Event 'High Limit Value', Status 'Acknowledge', Info 'Limit 0.1%', Comment 'High Limit Value'. Row 4: Date 11/11/2016, Time 7:45:15, Priority 0, WinCC message text 'Low Limit Value', Event 'Low Limit Value', Status 'Acknowledge', Info 'Limit 0.1%', Comment 'Low Limit Value'. Rows 5 to 31 are empty. The bottom status bar shows 'Pending: 5 / To acknowledge: 4 / Hidden: 0 / List: 4'.

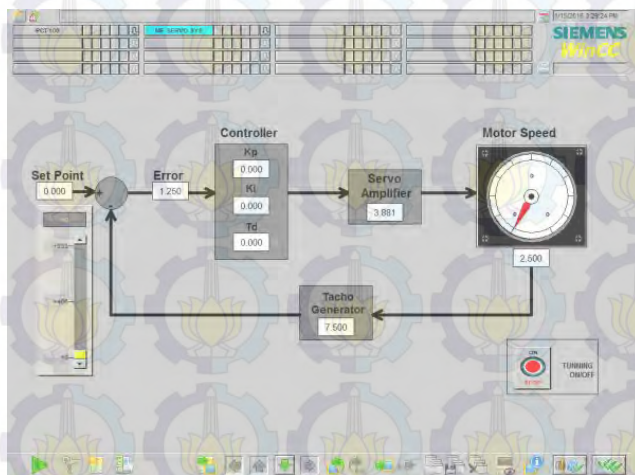
	Date	Time	Priority	WinCC message text	Event	Status	Info	Comment
1	11/11/2016	7:45:15	0	High Limit Value	High Limit Value	Acknowledge	Limit 0.1%	High Limit Value
2	11/11/2016	7:45:15	0	High Limit Value	High Limit Value	Acknowledge	Limit 0.1%	High Limit Value
3	11/11/2016	7:45:15	0	High Limit Value	High Limit Value	Acknowledge	Limit 0.1%	High Limit Value
4	11/11/2016	7:45:15	0	Low Limit Value	Low Limit Value	Acknowledge	Limit 0.1%	Low Limit Value
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								

At the bottom of the window, the status bar displays: Pending: 5 / To acknowledge: 4 / Hidden: 0 / List: 4.

The screenshot displays the Siemens WinCC Graphics Designer interface. At the top, the title bar shows the date and time: 11/10/2016 7:45:19 PM. The main window is titled 'incoming alarm list'. Below the title bar, there is a toolbar with various icons for editing and viewing. The central area contains a table with the following columns: Date, Time, Priority, WinCC message text, Event, Status, Info, and Comment. The table has four rows of data, each with a yellow background. The first row shows a 'High Limit Value' alarm, the second row shows a 'High Limit Value' alarm, the third row shows a 'High Limit Value' alarm, and the fourth row shows a 'Low Limit Value' alarm. The background of the main window features a large gear and lotus flower graphic. The bottom status bar shows 'Pending 5', 'No acknowledgement: 4', 'Hidden 0', and 'List: 4'.

Date	Time	Priority	WinCC message text	Event	Status	Info	Comment
1		0	High Limit Value				
2		0	High Limit Value				
3		0	High Limit Value				
4		0	Low Limit Value				

5. Motor DC home screen





DAFTAR PUSTAKA

- [1] Doris Saez; Aldo Cipriano; and Andrzej W. Ordys, “Optimisation of Industrial Processes at Supervisory Level”, Springer-Verlag, London, 2002.
- [2] B. Bulut; M.R. Katebi; M.A. Johnson, “Industrial Application of Model Based Predictive Control as a Supervisory System”, Proceedings of the American Control Conference, Chicago, Illinois, June, 2000.
- [3] Med. Essahafi, “Model Predictive Control (MPC) Applied to Coupled Tank Liquid Level System”, Laboratory of Autmation and Energy (LACE) Faculty of Science and Technology-CED-Science and Technology, 2014.
- [4] M. Sbarciog; R. De Keyser; S. Cristea; and C. De Prada, “Nonlinear predictive control of processes with variable time delay. A temperature control case study”, 17th IEEE International Conference on Control Applications Part of 2008 IEEE Multi-conference on Systems and Control San Antonio, Texas, USA, September 3-5, 2008.
- [5] M. D. Mufti; S. J. Iqbal; S. A. Lone; and Qurat-ul Ain, “Supervisory Adaptive Predictive Control Scheme for Supercapacitor Energy Storage System”, IEEE, October 17, 2013.
- [6] Wei QI; Jinfeng Liu; Xianzhong Chen; and Panagiotis D, “Supervisory Predictive Control of Standalone Wind/Solar Energy Generation Systems”, IEEE Transactions on Control Systems Technology, Vol. 19, No. 1, January, 2011.
- [7] Guolian Hou; Jingbin Liu; Pengcheng Jiang; Jianhua Zhang; “ The Application of Supervisory Predictive Control in Supercritical Unit Coordinated Control System”, IEEE 8th Conference of Industrial Electronics and Applications (ICIEA), Melbourne, Australia, 2013.
- [8] Normey-Rico, J.E., “Control of Dead-time Processes”, Springer, XXV, 462, 2007.
- [9] Rames. C. Panda; Cheng-Ching Yu; Hsiao-Ping Huang; “PID Tuning Rules for SOPDT Systems: Review and Some New Results”, ISA Transactions 43 (283-295), The Instrumentation, Systems, and Automation Society, Taiwan, 2004.
- [10] O. Katsuhiko, “Modern Control Engineering, 5th edition”, Prentice Hall, United States, 2010.
- [11] Karl J. Astrom and Tore Hagglund, “PID Controllers. 2nd Edition”, Instrument Soociety of America, 1995.

- [12], "Process Control Technology PCT-100 manual" BYTRONIC Educational Technology, Staffordshire, England.
- [13] Kenny Uren, "Predictive PID Control of Non-Minimum Phase Systems". South Africa. 2011.
- [15] Messner, Bill ; Down Tilbury, "Introduction PID Control Design", <http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction§ion=ControlPID>, 2011.
- [16] LiuPing Wang, PhD, "Model Predictive Control System Design and Implementation Using", Springer-Verlag, London, 2009.
- [17] Mirzal Andri, "Stability Analysis and Compensation of Time Delays in Analog Control Systems", International Journal of Control and Automation Vol. 5, No. 4, December, 2012.

RIWAYAT HIDUP



Riki Rizki lahir di Takengon pada tanggal 19 September 1994. Penulis telah menyelesaikan pendidikannya di TK YKA Banda Aceh, SDN 20 Kota Banda Aceh dan SDN 9 Kota Tapaktuan, SMPN 1 Tapaktuan, dan SMAN Unggul Kabupaten Aceh Selatan. Pada tahun 2012, penulis melanjutkan studi di Institut Teknologi Sepuluh Nopember (ITS), Jurusan Teknik Elektro melalui Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN) jalur undangan. Di Jurusan Teknik Elektro ini, penulis memilih Teknik Sistem Pengaturan sebagai bidang studi yang ditekuni. Pada bulan Januari 2016, penulis mengikuti

seminar dan ujian Tugas Akhir sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik Elektro.